Geospatial Metadata for Discovery in Scholarly Publishing

Bachelor Thesis

Institute for Geoinformatics
University of Münster

Author:	Tom Niers
Supervisor:	Dipl-Geoinf. Daniel Nüst
Co-supervisor:	Prof. Dr. Edzer Pebesma

Münster, October 2020

Abstract

Many scientific articles are related to specific regions of the Earth. The connection is often implicit, although geospatial metadata has been shown to have positive effects, such as detecting biases in research coverage or enhancing discovery of research. Scholarly communication platforms lack an explicit modeling of geospatial metadata. In this work, we report a novel approach to integrate well-defined geospatial metadata into Open Journal Systems (OJS). Authors can create complex geometries to represent the related location(s) or region(s) for their submission and define the relevant time period. They are assisted by an interactive map and a gazetteer to capture high quality coordinates as well as a matching textual description with high usability. The geospatial metadata is published within the article pages using semantic tags, integrated in standardized publication metadata, and shown on maps. Thereby, the *geoOJS* plugin facilitates indexing by search engines, can improve accessibility, and provides a foundation for more powerful map-based discovery of research articles across journals.

Table of Contents

List of Abbreviations **IV**

List of Figures **V**

1 Introduction **1**

2 Related Work **2**

2.1 Existing Approaches to Integrate Geospatial Metadata for Discovery in Scholarly Publishing Outside of OJS 2

2.2 Existing Approaches to Integrate Geospatial Metadata for Discovery in OJS 3

3 Geospatial Metadata for Discovery in Scholarly Publishing **4**

3.1 Potential of Integrating Geospatial Metadata in Scholarly Publishing 4

3.2 Collection of Geospatial Metadata for Scientific Articles 5

3.2.1 Achieving Geospatial Metadata by Analyzing the Scientific Article 5

3.2.2 Achieving Geospatial Metadata by an Author Input 7

3.3 Standardization of Geospatial Metadata for Scientific Articles in Scholarly Publishing 9

3.4 Storage of Geospatial Metadata in Scholarly Publishing 11

3.5 Validation of Geospatial Metadata in Scholarly Publishing 12

3.6 Accessibility of Geospatial Metadata in Scholarly Publishing 12

3.7 Requirements 13

3.7.1 Requirements Concerning the Potential of Integrating Geospatial Metadata in Scholarly Publishing 13

3.7.2 Requirements Concerning the Collection of Geospatial Metadata for Scientific Articles 14

3.7.3 Requirements Concerning the Standardization of Geospatial Metadata for Scientific Articles in Scholarly Publishing 15

3.7.4 Requirements Concerning the Storage of Geospatial Metadata in Scholarly Publishing 16

3.7.5 Requirements Concerning the Validation of Geospatial Metadata in Scholarly Publishing 17

3.7.6 Requirements Concerning the Accessibility of Geospatial Metadata in Scholarly Publishing 17

4 *geoOJS* – Well-Defined Geospatial Metadata for Discovery in OJS **18**

4.1 Conceptual Approach of Requirements for *geoOJS* 18

4.1.1 Collection of Well-Defined Geospatial Metadata in OJS 18

4.1.2 Geospatial Metadata for Discovery of Scientific Articles in OJS 21

4.2 The *geoOJS* Plugin 23

4.2.1 Architecture of the *geoOJS* Plugin 23

4.2.2 Implementation of the *geoOJS* Plugin 24

4.2.3 Dependencies Used for the *geoOJS* Plugin 27

4.3 Evaluation of *geoOJS* Concerning the Requirements 29

5 Conclusion **31**

References **33**

Appendix **34**

A Examples for Storage of Geospatial Metadata in the OJS Database 34
A.1 Example for the Temporal Properties in the OJS Database 34
A.2 Example for the Spatial Properties in the OJS Database 34
A.3 Example for the Coverage Information in the OJS Database 38
B Examples for Standardized Storage of Geospatial Metadata in the HTML Document Header of the OJS Article View 39
B.1 Example for a Geo Meta Tag in the HTML Document Header of the OJS Article View 39
B.2 Example for DCMI Box Encoding Scheme in the HTML Document Header of the OJS Article View 39
B.3 Example for ISO 19139 in the HTML Document Header of the OJS Article View 39
B.4 Example for ISO 8601 in the HTML Document Header of the OJS Article View 39
C Screenshots of *geoOJS* 40
C.1 Screenshot of *geoOJS* – Input of Geospatial Metadata 40
C.2 Screenshot of *geoOJS* – Geospatial Properties in the OJS Article View . . . 41

Declaration of Academic Integrity **42**

List of Abbreviations

API Application Programming Interface. 2, 12

CDN Content Delivery Network. 24

CSS Cascading Style Sheets. 24

CSV Comma-separated values. 7

CSW Web Catalogue Service. 13

GeoJSON Geographic JSON. 7, 11, 12, 21, 22, 25, 26, 29

GeoTIFF Geographic Tagged Image File Format. 7

GFZ German Research Centre for Geosciences. 13

GLOBE Global Collaboration Engine. 2

GMT Greenwich Mean Time. 19, 21, 29

GSW GeoScienceWorld. 2

HTML Hypertext Markup Language. V, 13, 17, 22, 23, 25, 27, 31

ISO International Organization for Standardization. 11, 13, 22

JS JavaScript. 26

JSON JavaScript Object Notation. 26

KML Keyhole Markup Language. 2, 7

OGC Open Geospatial Consortium. 13

OJS Open Journal Systems. I, II, 3–5, 11, 13, 14, 18–31

PANGEA Data Publisher for Earth & Environmental Science. 2, 13

PKP Public Knowledge Team. I, 23, 26

REST Representational State Transfer. 2

SQL Structured Query Language. 11, 25

UTC Coordinated Universal Time. 11, 19

WCS Web Coverage Service. 13

WFS Web Feature Service. 6, 13

List of Figures

3.1 Mockup – Autocompletion for Spatial Data in a Form 8
4.1 Screenshot of *geoOJS* – Submission Process 18
4.2 Screenshot of *geoOJS* – Article View 22
4.3 HTML Source Code Generated by *geoOJS* – Geospatial Metadata in the HTML Document Header of the OJS Article View 23
4.4 The *geoOJS* Plugin 24
4.5 Interaction Between Different Parts of the *geoOJS* Plugin 25
4.6 Geospatial Metadata in the OJS Database 26
4.7 Storage and Visualization Chain of the *geoOJS* Plugin 27
4.8 Screenshot of *geoOJS* – Plugin Settings 28

1 Introduction

Almost every scientific article that refers to existing regions of the earth, contains "[...] a narrative description of the study area" (Karl, 2019). At the same time, Shapiro and Báldi (2012) found that more than a quarter of articles omit maps and coordinates and only use vague descriptions albeit the relevance of location for said article's content. Geospatial metadata can help to detect biases in research coverage (Karl, Herrick, et al., 2013; Young & Lutters, 2017), to filter search results for scientific articles (Howell et al., 2019; Karl, 2019; Karl, Herrick, et al., 2013), and to enhance the understanding of relations within a study area (Margulies, Magliocca, Schmill, & Ellis, 2016). In more than half of the scientific articles that refer to locations, coordinates are used to determine the location (Karl, Herrick, et al., 2013; Shapiro & Báldi, 2012). However, geospatial information on scientific articles is not yet exploited in scholarly publishing platforms. Coordinates can be included in articles in different formats (Karl, 2019; Kmoch, Uuemaa, Klug, & Cameron, 2018) and are therefore prone to errors such as improper formatting, incompleteness, and ambiguity (Karl, 2019; Margulies et al., 2016), so that the demand for standardization increases (Karl, 2019; Karl, Herrick, et al., 2013; Kmoch et al., 2018; Margulies et al., 2016; Young & Lutters, 2017).

In this work, we discuss the challenges and potential in the acquisition of well-defined geospatial metadata. We report on a novel approach to integrate well-defined geospatial metadata in a scholarly publishing platform to enhance discovery of scientific articles. For this purpose we develop the prototype *geoOJS*, which offers a novel way for authors to provide spatial properties of research works when submitting an article to a journal based on the open source software OJS[1].

OJS is an open source software tool for scholarly publishing developed by PKP[2]. It is the most widely used open source journal publishing platform. The entire workflow for scholarly publishing is covered flexibly and extensibly. For this purpose plugins can extend the system as desired and add new features. Our considerations mentioned above are implemented by a plugin named *geoOJS* as prototype for OJS.

Research Questions

This thesis aims to address the following research questions:

1. How can geospatial metadata be integrated in scholarly publishing platforms so that it can be used for discovery of scientific articles?
    (a) How can geospatial metadata be stored and shared in public databases of scientific articles?
    (b) How can geospatial metadata help to put scientific articles in relation?
2. Which means of publication are most suitable for geospatial metadata to achieve accessibility for humans and machines?

[1] `https://pkp.sfu.ca/ojs/` (Retr. 2020-10-09)
[2] `https://pkp.sfu.ca/` (Retr. 2020-10-09)

2 Related Work

2.1 Existing Approaches to Integrate Geospatial Metadata for Discovery in Scholarly Publishing Outside of OJS

GLOBE[3] is an online environment that allows researchers to compare studies. It is focused on the exchange of research work in the field of land change systems. GLOBE enables the search for study cases using a geographic extent.[4] With an interactive map and a configurable bounding box, the user can choose for which area results should be shown. The geospatial metadata can be defined in different ways: by uploading the data as KML[5] or Shapefile[6] , creating it by drawing, or selecting a predefined spatial entity from a data layer.[7] In the search view the case studies are only displayed as markers, in the single view of the case studies more precisely as polygons.

JournalMap[8] is a scientific literature search engine which provides the possibility to browse scientific articles by location. Using a world map, the user has the possibility to zoom in on the area of interest.[9] Then only the scientific articles which are assigned to this area will be displayed. The required geospatial metadata were retrieved from the scientific articles. Coordinates are extracted if available, standardized to a uniform format, and corrected in case of errors. If no coordinates are available, the location is determined manually. However, the scientific articles are only represented by single coordinates, although it is planned to support bounding boxes and polygons in the future.[10] The published data, including location information, can be accessed via a broad REST API which is freely available upon registration.[11]

PANGEA[12] is an information system operating as an Open Access library with the goal to store georeferenced data for earth system research. This system is particularly suitable for the storage of data sets. Thereby geospatial metadata can be used as parameters for the search. Besides coordinate input, a drawn bounding box or a time period can be applied to the search. The geospatial metadata are submitted in the form of events[13], including coordinates, as part of the submission.

GSW[14], a resource for scientific work in the Earth Sciences, allows to limit search in scholarly publishing by geographic location through the integration of OpenGeoSci Map[15]. A drawn bounding box, coordinates and the time period can be used to limit the data. The data basis for this purpose is data from GeoRef[16].

[3] `http://globe.umbc.edu` (Retr. 2020-10-07)
[4] GLOBE map (registration required): `http://globe.umbc.edu/app/#/manager` (Retr. 2020-10-07)
[5] `https://www.ogc.org/standards/kml` (Retr. 2020-09-28)
[6] `https://desktop.arcgis.com/en/arcmap/latest/manage-data/shapefiles/what-is-a-shapefile.htm` (Retr. 2020-09-28)
[7] GLOBE case creation: `http://globe.umbc.edu/tutorials/case-creation-procedure/` (Retr. 2020-10-07)
[8] `https://www.journalmap.org/` (Retr. 2020-10-07)
[9] JournalMap search: `https://www.journalmap.org/search#list?bounds=66.79191,174.37500|-58.44773,129.37500&precision=1` (Retr. 2020-10-07)
[10] JournalMap idea: `https://www.journalmap.org/about` (Retr. 2020-10-07)
[11] `https://www.journalmap.org/developer/documentation/1-0` (Retr. 2020-10-07)
[12] `https://www.pangaea.de/` (Retr. 2020-10-07)
[13] PANGEA event: `https://wiki.pangaea.de/wiki/Event` (Retr. 2020-10-07)
[14] `https://pubs.geoscienceworld.org/` (Retr. 2020-10-07)
[15] `https://pubs.geoscienceworld.org/pages/opengeosci` (Retr. 2020-10-07)
[16] Large geoscience literature database (`https://pubs.geoscienceworld.org/georef` (Retr. 2020-10-07))

2.2 Existing Approaches to Integrate Geospatial Metadata for Discovery in OJS

By enabling the Usage Statistics Plugin[17], regional data of the reader including country, region and city information can be tracked. It can be determined which users visit the article pages or download the article ordered by country, region, city, month, and day. However, this is not geospatial metadata concerning the content of the scientific article, but rather user statistics in terms of time and location.

Besides, there is the Dublin Core 1.1 meta-data Plugin[18], which is automatically integrated into OJS. This plugin allows to store additional meta data that complies to the Dublin Core Standard[19]. In the OJS settings additional metadata fields can be activated, which the author then can specify in the submission process. One of these fields is the coverage information. In OJS the coverage information is described as follows: "Coverage will typically indicate a work's spatial location (a place name or geographic coordinates), temporal period (a period label, date, or date range) or jurisdiction (such as a named administrative entity)".[20] This corresponds to the coverage element[21] of the Dublin Core Metadata Initiative. The user has the possibility to define geospatial properties for the content of the scientific article, as far as coverage information is activated as metadata element.[22]

[17] `https://docs.pkp.sfu.ca/admin-guide/en/statistics#configure-regional-data-tracking` (Retr. 2020-10-08)
[18] `https://github.com/pkp/ojs/tree/master/plugins/generic/dublinCoreMeta` (Retr. 2020-10-08)
[19] `https://www.dublincore.org/specifications/dublin-core/dcmi-terms/` (Retr. 2020-10-08)
[20] Description for coverage information in the workflow settings of OJS (`https://github.com/pkp/ojs` (Retr. 2020-10-09))
[21] `https://www.dublincore.org/specifications/dublin-core/coverage-element/` (Retr. 2020-10-08)
[22] Activation of coverage information in OJS: `https://github.com/pkp/ojs/blob/46c97d55f0b346bb83dd542cea5384c057c36b6b/plugins/generic/dublinCoreMeta/DublinCoreMetaPlugin.inc.php#L70` (Retr. 2020-10-08)

3 Geospatial Metadata for Discovery in Scholarly Publishing

Geospatial metadata can be of great value for discovery in scholarly publishing. In this chapter we discuss the potential of geospatial metadata as well as the steps necessary to make them available. To make geospatial metadata usable for the discovery of scientific articles, different aspects have to be considered. First of all, the data must be collected. Second, there is a need to standardize the data in order to achieve comparability between different articles. Third, the data has to be stored adequately to make them in many different ways useable and accessible for a broad user group. Fourth, a validation is worth considering to verify the validity of the data.

We present the different approaches and describe the resulting requirements in the form of user stories. To limit the level of abstraction we relate our considerations to OJS. Nevertheless the considerations are generally transferable to other scholarly publishing systems.

3.1 Potential of Integrating Geospatial Metadata in Scholarly Publishing

First, geospatial metadata can help to detect biases in research coverage. It can be determined which areas of investigation occur disproportionately often and which are neglected (Karl, Herrick, et al., 2013; Martin, Blossey, & Ellis, 2012). This knowledge can be used as an additional factor in the selection of future study areas to avoid redundancy (Young & Lutters, 2017). On the contrary, gaps in research can be eliminated (Hendrickx, Dujardin, Pickering, & Alvar, 2010; Howell et al., 2019).

Second, correlations can become visible, which otherwise would not be noticed. If one recognizes similar behavior in different studies based on the location, one can possibly transfer results or even the used methods (Karl, Herrick, et al., 2013).

Third, search results for scientific articles can be filtered by integrating geospatial filters. Most of today's search engines for scientific articles are thematically based and leave out geographical data for filtering. The integration of geographical data into the search filter would increase the hits for potential interesting articles (Karl, Gillan, & Herrick, 2013). Karl, Herrick, et al. (2013) discussed how the exact location integrated in a geosemantic search can contribute to improve the search for scientific articles. Literature search tools that include geospatial data in their queries have high potential to become the preferred search engines in research (Howell et al., 2019).

Fourth, geospatial metadata can enhance the understanding of relations within a study area. Thus, both the regional and global context of case study becomes knowable (Margulies et al., 2016). The availability of comprehensive geospatial metadata can facilitate reuse and citation of published work.

Fifth, geospatial data enables creation of maps. Interactive maps provide authors with another way of presenting the article's spatial properties. The user is able to gain deeper insights and achieve a broader overview over the data. Effectiveness of user's research exploration is increasing (Elsevier, n.d.). Visualization of geospatial metadata in a map allows the reader to decide whether an article and the corresponding data are of interest or not. Young and Lutters (2017) found that researchers understand data best when they are presented within their geospatial context. A map is useful in two ways, in the search for data and in the process of understanding data. The map representation and corresponding storage of coordinates can lead to the elimination of ambiguity. If a scientific work deals with a place like "Münster", which exists e.g. in Germany independently of each other in three different federal states, a map can provide unambiguous clarity.

The multitude of possibilities that geospatial metadata offers leads to demands on journals and publishers to provide geospatial metadata in a default manner. Geospatial metadata should be added to scientific articles, so that the potential can be exhausted. Journals, publisher, and database manager, should actively request the storage of location data (Kmoch et al., 2018). This should not only apply to future scientific work, but also to articles that have already been published (Karl, Herrick, et al., 2013). There must be general clarity about the possibilities of storing geospatial metadata, so that there are almost no excuses for not storing geospatial properties of the article's content ("A place for everything", 2008).

At first sight the potential mentioned above certainly appeals to scientists in whose research locations are of high relevance. Accordingly, the target group, which could benefit from the integration of geospatial metadata in scholarly publishing, primarily includes scientific works from the natural science context. But, any scientific work that refers to a location can benefit, whether it is specifically about a single location that is being considered, different places and their comparison or a historical comparison. Space and time are, just like thematic content, crucial dimensions to describe information (Scheider, Degbelo, Kuhn, & Przibytzin, 2014). In other words, there is a large number of stakeholders for whom the use of geospatial metadata is worthwhile: at a scholarly publishing system like OJS, there are different types of users like authors who write articles, readers who read articles, administrators who manage journals and publishers of journals. These are the users to whom we relate our requirements in the form of user stories.

3.2 Collection of Geospatial Metadata for Scientific Articles

In order to be able to use geospatial metadata in the context of scholarly publishing, they must first be available or recorded. There are two main opportunities to collect geospatial data of scientific articles. On the one hand, the scientific article, i.e. the text, contained figures and datasets can be used as data basis. An examination of the scientific article can capture information about location and time of the article's content. On the other hand, the data can become available by other means without analyzing the article. The author can add the geospatial metadata himself by an input, or also in form of a further file upload. This chapter present approaches of data collection for geospatial metadata.

3.2.1 Achieving Geospatial Metadata by Analyzing the Scientific Article

One approach for collecting geospatial metadata of a scientific article is to generate the data from the finished article itself.

First of all it is possible to manually examine existing scientific articles for coordinates. Howell et al. (2019) have developed an online tool[23] with a corresponding database, which specifies location of publications and displays them on a map. Scientific articles such as journal articles, dissertations, and books were searched e.g. by using Google Scholar and examined for their spatial properties. If available, coordinates e.g. found in the methods section of the article were saved as a location. Otherwise, if coordinates were not available, the location was estimated manually using

[23] Aspen Ecology and Management – publication distribution on a map: `https://goo.gl/hyT6QR` (Retr. 2020-09-28)

Google Earth and expert knowledge. In addition to the usual metadata such as title, author, year, the location was then assigned to the scientific articles metadata, both textually and in form of coordinates (latitude and longitude in decimal degrees).

Second, scientific articles can be automatically examined for geographical coordinates to geotag them. Karl (2019) showed how pattern-matching algorithms, also known as geoparser, can be used to identify geographical coordinates and extract them from the text of the scientific article. Two types of geoparsers have been developed, one based on regular expressions[24] and the other one on lexical parsing. A regular expression parser compares a string, in this case the scientific article, with a pattern definition, in this case a coordinate format, and extracts all text snippets that match the pattern definition. A lexical parser splits the text into individual parts (e.g. words, phrases and other meaningful elements) and then parses this parts according to the coordinate parsers grammar syntax. In comparison, regular expressions are faster but much more difficult to read when using complex patterns compared to lexical parsing. Karl (2019) suggests a hybrid model which exploits the strengths of both models. If a simplified regular expressions parser does not find coordinates in the article, it is re-examined by the more robust lexical parser.

Third, the different components of scientific articles (title, abstract, full text) can be examined for place names, based on text recognition. Kmoch et al. (2018) showed how to use a gazetteer[25], as WFS service[26] to compare it with the text of scientific articles. For efficiency reasons the gazetteer data set was downloaded and a direct text-matching strategy matched elements from text and gazetteer. Regular expressions were used to ignore matches of too many irrelevant partial expressions. Eventually a list of the locations contained in the scientific article was retrieved and stored in an Excel spreadsheet.

Fourth, it can be checked whether a scientific article contains maps, which can then provide information about the location of the article's content. The figures contained in the text are examined for maps and the associated location-based information. Mohr (2017) demonstrated how maps can be recognized and how the contained axes, coordinates and names can be used to receive information about the spatial properties.

As shown there are several ways to examine a scientific article for geospatial properties. It need not be said that the first option presented, a manual analysis, is time-consuming compared to the others, as the text must be read and checked completely manually. But the analysis by a human being allows to expect a higher validity, compared to an automatic approach which attempts to recognize place names. Scientific articles often contain irrelevant place names (Karl, 2019). It can also happen that instead of the place name the name of a person was meant and nevertheless the name is interpreted as place name. Also, the place can exist more often, which makes the assignment more difficult. This difficulty can be solved with a validation by the author, e.g. the author accepts, edits or denies the result of text analysis. However, this validation would have to be integrated into the publication process. A subsequent validation after publication and the associated process is not feasible at all, or if possible, time-consuming. To that effect, the approach of examining geographic coordinates is more precise. The methods performed by Karl (2019) were correct in more than 85% of cases. Nevertheless there are also error sources for geoparsers, e.g.

[24] `https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_Expressions` (Retr. 2020-09-28)
[25] A register for locations.
[26] `https://www.ogc.org/standards/wfs` (Retr. 2020-09-28)

bounding boxes – if a geoparser detects one, but is actually expecting single pairs of coordinates, it may become confused.

Obviously, the choice of the best approach always depends on the components contained in the scientific article. If one can assume a certain quality and accuracy of the automatic approaches, they are the best way to add geospatial metadata to already written scientific papers. To do this manually would not be feasible for the large number of scientific articles that have been already published. However, a broad database of scientific articles with geospatial metadata is necessary. Automatic approaches can be the solution to unleash the potential described in chapter 3.1.

3.2.2 Achieving Geospatial Metadata by an Author Input

Besides approaches of analyzing the article, geospatial properties can also be identified by procedures that require further action by a human.

A conceivable solution would be a file upload with geospatial metadata stored in it. This can be a file like GeoJSON[27], KML[28] or a Shapefile which gives explicit information about the geospatial properties (Konkol & Kray, 2019; Magliocca et al., 2015; Shapiro & Báldi, 2012; Young & Lutters, 2017). These can, for example, be included in the appendix of the scientific article (Shapiro & Báldi, 2012). Nüst and Qamaz (2020) have developed a library for extracting the geospatial extent from data files. Using this, it is possible to extract a time interval and a bounding box from GeoJSON, CSV[29], GeoTIFF[30] or Shapefiles. A further file for geospatial metadata can be viewed separately from the scientific article, as an additional upload, or as part of the scientific article. It might be worth considering to make it a usual practice to add a file to the scientific article that explicitly describes the geospatial metadata.

Another option is that the author who is publishing the article creates the geospatial metadata himself e.g. during submission of the article. The author defines the place to which the scientific article refers to by providing a textual description, e.g. a city name. An autocomplete[31] can be used to provide the author with suggestions for the input, this simplifies the author's work. Figure 3.1 shows a mockup how this could be implemented. The user would like to save `Muenster` and after entering the partial word `Muen` there are already various suggestions.

A further possibility is that the author creates the location by drawing a geometric shape on a map. A choice of different geometric shapes is useful to respond to different requirements:

- If the author wants to represent an explicit point, a marker is best suited.
- If the author wants to represent roughly the area of e.g. a city, a bounding box is useful.
- If the author wants to represent an area as exact as possible, a polygon with a not limited number of points should be available.
- If the author wants to represent a certain distance or track, a polyline is useful.

The author should have the possibility to insert multiple geometric shapes, since a study can contain several locations, e.g. if it is a comparison. Therefore there should be no limit to the number of geometric shapes.

[27] `https://geojson.org/` (Retr. 2020-09-28)
[28] `https://www.ogc.org/standards/kml` (Retr. 2020-09-28)
[29] `https://en.wikipedia.org/wiki/Comma-separated_values` (Retr. 2020-09-29)
[30] `https://www.ogc.org/standards/geotiff` (Retr. 2020-09-29)
[31] `https://www.w3schools.com/tags/att_input_autocomplete.asp` (Retr. 2020-09-29)

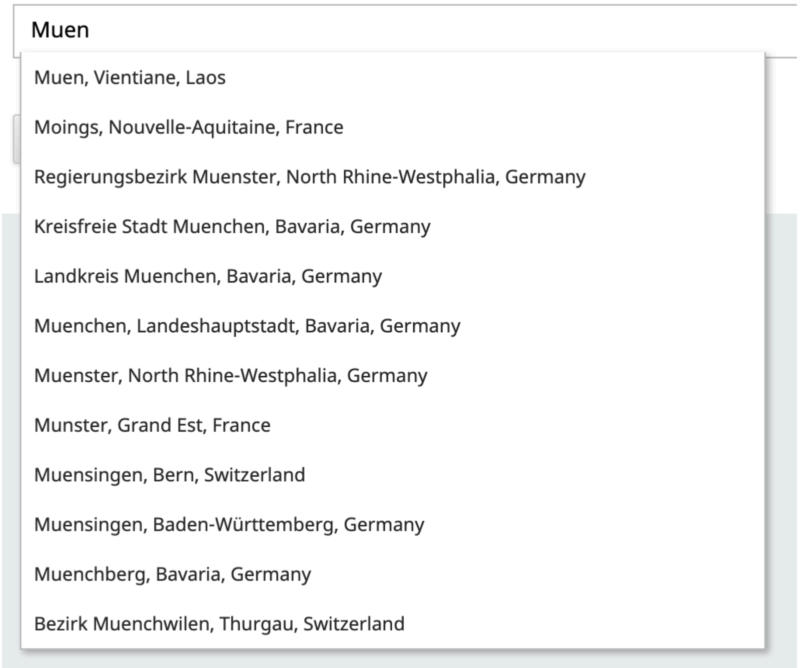

Figure 3.1: Mockup – input of spatial data in a form, by use of autocompletion

An integrated search which is also based on autocompletion can help the author to find locations. It can automatically suggest a geometric shape, which relieves the author of work. The author should have the ability to edit or delete his own input or any suggested bounding box, to avoid incorrect metadata.

In addition, the map should provide some further tools. A choice between different layers[32] is useful, depending on the author's field of research: sometimes a satellite image may be appropriate, in other cases a topographic map[33] may be more useful. For example, an urban planner can work well with a street map that shows the structure of cities and other man-made objects, while a landscape ecologist will probably prefer a satellite image in which landscape forms can be recognized. Just as well, other maps that emphasize certain topics can be useful. Accordingly, an opportunity for the author to add own layers would also be conceivable. Besides, it should be possible to change the map extent, so that the author can correctly assess the geometric shape(s) shown.

To define the temporal properties of the scientific article's content, the author should be able to define date and time. Depending on the research topic, time can play a relevant role. For an extreme weather event a precise time is perhaps more decisive in contrast to the observation of the behavior of living beings where time has to be recorded less exactly. It should be possible to enter the data as accurately as possible, but it should also not be required. An interactive widget, like a calendar view, can be used for quicker input of the author, as the sequence of year, month and week are directly visible. Attention must be paid to the time zone of the temporal properties input so that there are no inaccuracies between the author's and the reader's time zone.

[32] `https://support.esri.com/en/other-resources/gis-dictionary/term/ba1e96e7-4cae-4714-875a-a7e3488b8bb9` (Retr. 2020-09-29)

[33] `https://www.ga.gov.au/scientific-topics/national-location-information/topographic-maps-data/basics/what-is-a-topographic-map` (Retr. 2020-09-29)

Manual input of geospatial metadata by location or input forms is likely to require more effort from a human being than automated text analysis approaches, but compared to the analysis approach, the results are more likely to be valid.

3.3 Standardization of Geospatial Metadata for Scientific Articles in Scholarly Publishing

As already mentioned in chapter 3.2.1, geospatial data can appear in different types in scientific articles.

Spatial data can occur in text form e.g. in the form of place names, in the form of coordinates, as part of maps, or in data sets and corresponding files. In text form street names, city names, country names and other possibilities can be mentioned. The textual names can be in different languages. There is a multiplicity of map types that can describe geographical features. Coordinates can be found in a variety of different formats in scientific articles (Karl, 2019). They differ both in spelling with different formats and in the coordinate system to which they refer. Epsg.io[34] lists the different possible coordinate systems.[35] Moreover, there can always be differences in the precision of the spatial properties. If spatial data are included in a scientific article, then in various formats that use different standards, storage technologies, languages and nomenclatures (Kmoch et al., 2018).

Besides, errors can occur when specifying spatial data as described in chapter 3.2.1. Place names can be ambiguous due to duplication, or simply not meant to be place names. Coordinates can have errors due to incorrect inputs by the author or formatting and typesetting processes. The JournalMap[36] project found that approximately 2 to 5 % of studies reporting coordinates for location have obvious errors. This includes errors such as improper formatting, incomplete coordinates and invalide values. The actual number of errors is probably higher, as only obvious errors were recorded (Karl, 2019).

In the same way, the temporal properties of the article's content can refer to different time zones, which are also sensitive to errors.

The large range of different ways to describe geospatial properties of scientific articles is prone to errors. This has the effect that the demand for standardization increases. One of the reasons for the lack of spatial properties is that standards and requirements for the integration of geospatial metadata are not provided. Thus, information is lost if it is not stored in a usable form. This has the consequence that the potential described in chapter 3.1 cannot be exhausted (Karl, Herrick, et al., 2013). Therefore there is the demand for uniform standards, guidelines, and conventions (Karl, 2019; Margulies et al., 2016). The question arises how to best establish this agreement of standardization.

First, Karl (2019) has developed principles that should be considered when defining standards for geospatial metadata of scientific articles. The standards should be complete, flexible, extensible, machine readable, and not confusing, and the location information should be provided at an appropriate scale.

Second, there is the possibility to use geospatial properties which have already been recorded, and parse them into clearly defined geographic spatial entities. Depending on the use case, other

[34] `https://epsg.io/` (Retr. 2020-09-27)
[35] Selection of coordinate reference systems: `https://epsg.io/?q=` (Retr. 2020-09-27)
[36] `https://www.journalmap.org/` (Retr. 2020-09-27)

spatial entities can be selected. World Protected Areas[37] are probably more interesting for landscape ecologists, while postal codes are more worthwile for economists (Young & Lutters, 2017). In this context it is reasonable to think about the use of geographical norm data[38], e.g. matching the recorded geospatial properties to an administrative division. There is a globally valid system of geo-names[39] divided into different administrative divisions. These divisions are available in different dimension/levels.[40] The initially recorded data are assigned to the corresponding administrative division whichever suits best in terms of size and location. This way, data, and, thus, the scientific article gets much more comparable concerning its geospatial properties. In addition, it is possible that the exact storage of the study location is critical for cultural or economic reasons, and thus a standardized storage can solve this difficulty (Howell et al., 2019). However it must be considered that a non-standardized bounding box, which is not dependent to a certain area due to standards, can be more precise. Bounding boxes can contain areas that do not correspond to the actual study area (Karl, Herrick, et al., 2013). Therefore a storage of both the initially recorded data and the standardized data is conceivable (Margulies et al., 2016). This enables an optimized use of the data depending on the individual requirements of the user. On the one hand, the reader who is interested in the most exact allocation can be shown the area of interest by means of the initial recorded metadata on a map. On the other hand, the data are also stored in a standardized way to achieve a high degree of comparability. One has to keep in mind that the scope of application is limited to the user group that can assess and understand the corresponding spatial entities. Therefore it is reasonable that the author decides which scope the spatial entities come from. This way the spatial entities fit as well as possible to the research area and are therefore understandable for the reader. In this context, it may also be considered to use spatial entities based on the bottom up principle. Keßler, Maué, Heuer, and Bartoschek (2009) discussed an approach whereby the spatial entities were formed based on geotagged photos harvested from the web.

Third, there are already standards that can be used to store geospatial properties. The Dublin Core Metadata Terms[41] contains a coverage element. It is intended to describe geospatial properties for an object or a resource. Geometric shapes and textual descriptions as described in chapter 3.2.2 can be defined within this coverage element (Karl, 2019). This must be questioned critically, because the choice between different geometric shapes, temporal properties and text makes the coverage element ambiguous. There are other standards that are more explicit in this respect, e.g. a geo-tag[42], where the region, place name or position is defined, but where the content is clearly indicated by the name.

Here are some examples for the use of a geo-tag for the city of Graz:

```
<meta name="geo.region" content="AT-6" />

<meta name="geo.placename" content="Graz" />

<meta name="geo.position" content="47.0667;15.4500" />
```

[37] https://www.protectedplanet.net/en (Retr. 2020-09-27)
[38] https://www.dnb.de/DE/Professionell/Standardisierung/GND/gnd_node.html (Retr. 2020-09-27)
[39] https://www.geonames.org/export/codes.html (Retr. 2020-09-27)
[40] Choose any country to download a certain administrative level: https://gadm.org/download_country_v3.html (Retr. 2020-09-27)
[41] https://www.dublincore.org/specifications/dublin-core/dcmi-terms/ (Retr. 2020-09-30)
[42] https://de.wikipedia.org/wiki/Geo-Tag (Retr. 2020-09-27)

```
<meta name="ICBM" content="47.0667, 15.4500" />
```

There are also standards that define a bounding box such as the DCMI Box Encoding Scheme[43] and ISO 19139[44].

Fourth, the temporal properties should be stored in a standardized format and time zone. Regarding the format, a standardized storage as Unix Time Stamp[45] is useful. As it is a whole number or to be precise an `Integer`, the error rate is minimized. Concerning the time zone it is advisable to use the UTC[46], because it is the world time.

The ideas for standardization of geospatial metadata discussed here, are also relevant for the autosuggest considerations in chapter 3.2.2. Both autosuggest for a text input and for a search with a returned bounding box should return standardized data.

Finally, it must be stated that clear standards must be used to exploit the potential of storing geospatial properties. Otherwise, the variety of occurrence types and the associated errors lead to the fact that the potential of storing geospatial metadata cannot be fully exploited. It is obvious that spatial entities like administrative units are more comparable than explicit forms of study areas, however, the study area should also be described exactly, as this is also beneficial.

3.4 Storage of Geospatial Metadata in Scholarly Publishing

To exploit the potential of geospatial metadata described in chapter 3.1, the data must be stored accordingly.

Obviously, it is recommended to use standardized data for storage as described in chapter 3.3. If the data are already stored in the database in a standardized way, they are subsequently easier to use. This way data are more comprehensible than if they are saved in an individual format. As described in chapter 3.3, the data should still be stored as accurately as possible. Thus not standardized concerning the accuracy level of coordinates. However, still standardized regarding the storage format. For this purpose formats such as Shapefile, GeoJSON and KML are used by scientists to store geospatial data (Konkol & Kray, 2019; Young & Lutters, 2017). So it is recommended to store the coordinates in these formats. However, it is also conceivable to store more than just the coordinates in such a file, i.e. temporal properties and a suitable administrative unit. Then all data would be stored centrally and valid in one file. But one should consider that for a search or a filter over the data explicit fields in a database are necessary.

To save the data, a database must be available that is suitable for the storage of geospatial metadata. Therefore, depending on the individual system used for scholarly publishing, it is necessary to check if the already used database can store geospatial data or if another database has to be applied. OJS uses MySQL[47] or PostgreSQL[48], which are both capable of storing geospatial data. MongoDB[49], which is a common NoSQL[50] database, is also suitable for storing geospatial metadata. MongoDB supports query operations on geospatial data and provides corresponding

[43] `https://www.dublincore.org/specifications/dublin-core/dcmi-box/2005-07-25/` (Retr. 2020-09-27)
[44] `https://www.iso.org/standard/32557.html`, example: `https://boundingbox.klokantech.com/` (Retr. 2020-09-27)
[45] `https://www.unixtimestamp.com/` (Retr. 2020-09-27)
[46] `https://en.wikipedia.org/wiki/Coordinated_Universal_Time` (Retr. 2020-09-27)
[47] `https://www.mysql.com/` (Retr. 2020-09-28)
[48] `https://www.postgresql.org/` (Retr. 2020-09-28)
[49] `https://www.mongodb.com/` (Retr. 2020-09-28)
[50] `https://en.wikipedia.org/wiki/NoSQL` (Retr. 2020-09-28)

functions. With the help of valid GeoJSON or appropriately formatted coordinate pairs, the database can for example be searched for specific coordinates.[51]

It is reasonable to store the provenance[52] for each geospatial property. Such information is necessary for third parties to have trust in the origin of the data. In addition, a combination of different collection options can become very complex and thus it is possible to determine the basis of the data despite all this.

For the storage of geospatial data, it is recommended to store the data directly in a standardized way with regard to both the level of detail and the format. However, it is important to ensure that an appropriate database is available that is capable of storing geospatial data and that data are stored in such a way that the potential of the data can be exploited.

3.5 Validation of Geospatial Metadata in Scholarly Publishing

To achieve the highest possible credibility for the geospatial metadata, validation should be considered. The current publication and peer-review process is not adequate to guarantee reliable geospatial metadata (Karl, Gillan, & Herrick, 2013). Scheider et al. (2014) illustrated that there is a need to validate the achieved data. There are different possibilities to enable a validation.

First, the author can check the plausibility of the geospatial metadata himself by controlling them on a map and a corresponding view for the temporal properties. The geospatial metadata are displayed and the author can either accept or reject them.

Second, the geospatial metadata can also be displayed in a peer review process. In this case an editor checks if the geospatial metadata match the content of the article.

Third, it is feasible to use the automated possibilities presented in chapter 3.2.1. For example, if the author manually adds the input by drawing a geometric shape as described in chapter 3.2.2, this input can be checked by an automated procedure. The result of a text analysis for textual geographic properties or coordinates should match the input of the author.

There is a need to verify the validity of geospatial metadata. Choosing how to verify the data one has to keep in mind the error rate of automatic approaches, as already described in chapter 3.2.1. A validation by a human being is more likely to be correct. Hereby, validity checks as part of the peer review in the review process should be considered. Although peer review can be extended by this, a check by another person besides the author is quite reasonable.

3.6 Accessibility of Geospatial Metadata in Scholarly Publishing

In order to exploit the potentials mentioned in chapter 3.1, it is necessary that geospatial metadata are well accessible.

It is of great advantage to store geospatial metadata in a machine-readable way. In this way, geospatial metadata can be passed on automatically. For example, it is valuable to provide a file in which all geospatial metadata are stored in a standardized way, as described in chapter 3.4. If this file is then callable via an API, it can be further used by third parties. Retrieval by possible data

[51] Geospatial Queries in mongoDB: `https://docs.mongodb.com/manual/geospatial-queries/` (Retr. 2020-09-28)
[52] `https://www.w3.org/2005/Incubator/prov/wiki/What_Is_Provenance` (Retr. 2020-09-28)

repositories such as PANGEA or GFZ Data Services[53], for specific catalogues such as CKAN[54] or OGC CSW[55] and for geodata servers such as OGC WFS[56] or OGC WCS[57] is conceivable.

It is also useful to embed the standards listed in chapter 3.3 in the HTML document. By embedding metadata elements such as the geo-tag in the HTML header, they can be used for search engine optimization. It is controversial that search engines like Google rely on this geo-tag information,[58] anyway search engines that focus on finding scientific articles using geospatial filtering can deliberately capture the location of the article's content from a geotag.

It is also useful to provide standardized geospatial metadata as text in the HTML document. Not only because search engine operators like Google use the text embedded in the HTML to get information about location, but also because it is accessible for humans who depend on screen reading devices. The study location should therefore not only be shown visibly on map, but also in text form.[59] For the temporal properties there are also options for a standardized storage as metadata element in the HTML header e.g. ISO 8601[60]. In addition, color blindness and red-green weakness should also be considered, so that map and the elements displayed on it should be chosen with this knowledge.[61]

3.7 Requirements

The ideas and considerations discussed so far result in specific requirements regarding the use of geospatial metadata for discovery in scholarly publishing. Divided into the various subsections, we have defined the requirements in the form of user stories. Different roles were defined on the basis of the considerations regarding the target group in chapter 3.1. The user stories were created according to the system introduced by Schirmacher (n.d.). Each user story describes abstractly an activity with a certain goal which is done by a role. If the goal and the corresponding activity is obviously interesting for all roles, it was described only for the publisher to avoid duplication. The user stories are to a certain degree generally applicable, but especially suited for OJS.

The following roles are possible:

reader	A person who reads a scientific article.
author	A person who writes a scientific article.
administrator	A person who administrates a journal.
publisher	A person who publishes a journal.

3.7.1 Requirements Concerning the Potential of Integrating Geospatial Metadata in Scholarly Publishing

/US1/ As publisher I want that scientific articles published in my journal include geospatial metadata, so that discovery of scientific articles in my journal benefits.

[53] `https://dataservices.gfz-potsdam.de/` (Retr. 2020-10-07)
[54] `https://ckan.org/portfolio/geospatial/` (Retr. 2020-10-07)
[55] `https://www.ogc.org/standards/cat` (Retr. 2020-10-07)
[56] `https://www.ogc.org/standards/wfs` (Retr. 2020-10-07)
[57] `https://www.ogc.org/standards/wcs/` (Retr. 2020-10-07)
[58] Google and the geo-tag: `https://blog.seoprofiler.com/googles-geo-meta-tags-web-pages/` (Retr. 2020-09-29)
[59] Web Accessibility Guidelines concerning maps: `http://web-accessibility.carnegiemuseums.org/content/maps/` (Retr. 2020-09-29)
[60] `https://www.iso.org/iso-8601-date-and-time-format.html` (Retr. 2020-10-04)
[61] `https://sparkgeo.com/blog/the-accessibility-of-web-maps/` (Retr. 2020-09-29)

/US2/ As author I want geospatial metadata to be stored for my scientific article, so that geospatial properties of the article's content are displayed in the article view and thus contribute to the reader's understanding.

/US3/ As reader I want to use geospatial metadata as a filter in a system for scholarly publishing, so that I can find articles using geospatial properties and draw conclusions about the state of research.

/US4/ As reader I want to see all study areas on a map which appear in scientific articles of journals in a scholarly publication system, to be able to recognize spatial relationships between the articles.

/US5/ As reader I want the geospatial metadata of a scientific article to be displayed in the article view e.g. on a map, so that they help me to understand the geospatial context of the article.

3.7.2 Requirements Concerning the Collection of Geospatial Metadata for Scientific Articles

Concerning Achieving Geospatial Metadata by Analyzing the Scientific Article:

/US6/ As publisher I want that each scientific article is automatically analyzed for geospatial properties, no matter in which form they appear, so that geospatial metadata are available for each scientific article.

/US7/ As publisher I want that each scientific article is automatically analyzed for geospatial properties, so that geospatial metadata are available without the author having to enter it and a certain quality is ensured for the data despite the automation.

/US8/ As administrator I would like to automatically capture and save the geospatial metadata of all scientific articles of the journal by pushing a button in OJS, so that I have a minimum of workload but there are still geospatial metadata for all scientific articles available.

/US9/ As author I would like to have the geospatial metadata for my scientific article automatically stored, so that I do not have to do further steps and spend time on it.

Concerning Achieving Geospatial Metadata by an Author Input:

/US10/ As author I want to be able to upload geospatial metadata describing the content of the scientific article in a file, so that I can reuse them if I have already defined them.

/US11/ As author I want to define spatial properties by textual input and autosuggest, so that I get suggestions how to store them as metadata.

/US12/ As author I want to define spatial properties by drawing on a map with different geometric shapes, to represent the location as precisely as possible.

/US13/ As author I want to draw an unlimited number of geometric shapes on a map, so that I can also describe the spatial properties of scientific work that relates to several places.

/US14/ As author I would like a bounding box to be suggested as spatial property using a textual search with autosuggest, so that the effort for me is minimal.

/US15/ As author I want to choose between different layers for the map, so that I can adapt the map to my individual needs.

/US16/ As author I would like to add my own layers to the map, so that I can adapt the map to my individual needs.

/US17/ As author I want the map to contain possibilities to change the map extent, so that I can correctly estimate where my research area is located.

/US18/ As author I want to enter the temporal properties using an interactive widget, so that I can enter the correct time as quickly as possible.

/US19/ As author I want to define the time down to the second, so that the metadata are as accurate as possible.

/US20/ As author I want to enter the temporal properties in my time zone, so I do not have to convert the time.

/US21/ As reader I want the temporal properties to be displayed in my time zone, so I do not have to convert the time.

/US22/ As publisher I want the temporal properties to be saved in a default time zone, to avoid inaccuracies.

/US23/ As author I want to edit and delete geospatial properties also after they have been entered, in case they have been entered incorrectly.

*For the reader the user stories **/US14/**, **/US15/** and **/US17/** are applicable for the article view. With the restriction that user story **/US14/** is only a search, without a bounding box being suggested.*

3.7.3 Requirements Concerning the Standardization of Geospatial Metadata for Scientific Articles in Scholarly Publishing

/US24/ As publisher, I want the geospatial metadata to be stored in a standardized way so that no inaccuracies can occur and thus its potential can be optimally exploited.

/US25/ As author I want to have a suitable spatial entity automatically suggested to my textual input, so that the study area of my scientific work is comparable.

/US26/ As author I want to receive suggestions for the spatial entity that matches my textual input for the search, so that the study area of my scientific work is comparable.

/US27/ As author I want to have a suitable spatial entity automatically suggested for my geometric shape I have drawn, so that the study area of my scientific work is comparable.

/US28/ As author I want for each spatial entity that is suggested to me that the corresponding geometric shape is shown on a map, so that I can check if it really fits to my research area.

/US29/ As author I want to edit and delete every spatial entity that is suggested to me, so that it really fits to the study area and no wrong data are stored.

/US30/ As author I want to decide from which topic spatial entities should be proposed to me, so that they fit as well as possible to my research area.

/US31/ As author I want that if something is suggested to me, it does not happen without me noticing it, but that I am informed accordingly.

/US32/ As reader I want the standardized geospatial metadata to be displayed in the article view, so that I can get an impression of the scientific article's content regarding the geospatial properties and can compare it with other articles.

/US33/ As reader I want the spatial entity of each study location to be displayed, so that I can correctly classify the study location locally and globally.

3.7.4 Requirements Concerning the Storage of Geospatial Metadata in Scholarly Publishing

/US34/ As publisher I want the geospatial metadata to be stored in the database in a way, that its potential can be optimally exploited.

/US35/ As publisher I want the geospatial metadata to be stored in the database both as accurate as possible concerning the level of detail and standardized in terms of format and level of detail, so that their potential can be optimally exploited.

/US36/ As administrator I want the geospatial metadata to be stored in the corresponding database, so that they are easily filterable and searchable.

/US37/ As reader I want to download all geospatial metadata together in a single file, so that I can integrate them elsewhere and do not have a big effort.

/US38/ As administrator I want that for each geospatial property the provenance is stored, to be able to keep the track of how data were created.

3.7.5 Requirements Concerning the Validation of Geospatial Metadata in Scholarly Publishing

/US39/ As publisher I want the geospatial metadata to be validated, to ensure that only correct data are stored.

/US40/ As author I want to check the geospatial metadata myself, to make sure that they are correct and valid.

/US41/ As author I want the geospatial metadata to be checked, so that a second instance confirms the validity.

/US42/ As author I want my input to be automatically checked for validity, so I do not have to check it myself and still know that my input is valid.

3.7.6 Requirements Concerning the Accessibility of Geospatial Metadata in Scholarly Publishing

/US43/ As publisher I want the geospatial metadata to be accessible, so that they can be used for discovery.

/US44/ As publisher I want the geospatial metadata to be included in the HTML header of the scientific article as metadata, so that the articles can be found in a search based on the article's content geospatial properties.

/US45/ As publisher I want the geospatial metadata to follow accessibility standards, so that they can be captured by both, machines and humans.

/US46/ As reader who depends on screen reading devices, I want to collect information about the geospatial metadata of the scientific article through the text, so that I can capture this geospatial information by screen reading devices even if I cannot capture a map.

4 *geoOJS* – Well-Defined Geospatial Metadata for Discovery in OJS

An attempt was carried out how to achieve well-defined geospatial metadata and use them for discovery in scholarly publishing. For this purpose, a prototype for OJS was developed in compliance with the requirements mentioned in chapter 3.7. In this chapter we discuss the methodical realization of the requirements and describe the implementation. In order to provide an appropriate reference to the requirements, these are listed in brackets for methodical consideration.

4.1 Conceptual Approach of Requirements for *geoOJS*

The use of geospatial metadata in a system like OJS consists of two main parts. First, the collection of geospatial metadata, and second their use for discovery.

4.1.1 Collection of Well-Defined Geospatial Metadata in OJS

To add metadata to a scientific article, there is already a step "3rd Enter Metadata" during the submission process in OJS. In this step metadata like title and abstract are added. We decided to define geospatial metadata in this step as well, because it seems to be the right place for it (see Figure 4.1). Surely it would have been possible to define a new step for it, but since we want to use elements that already exist in this step to specialize them, it is the right place.

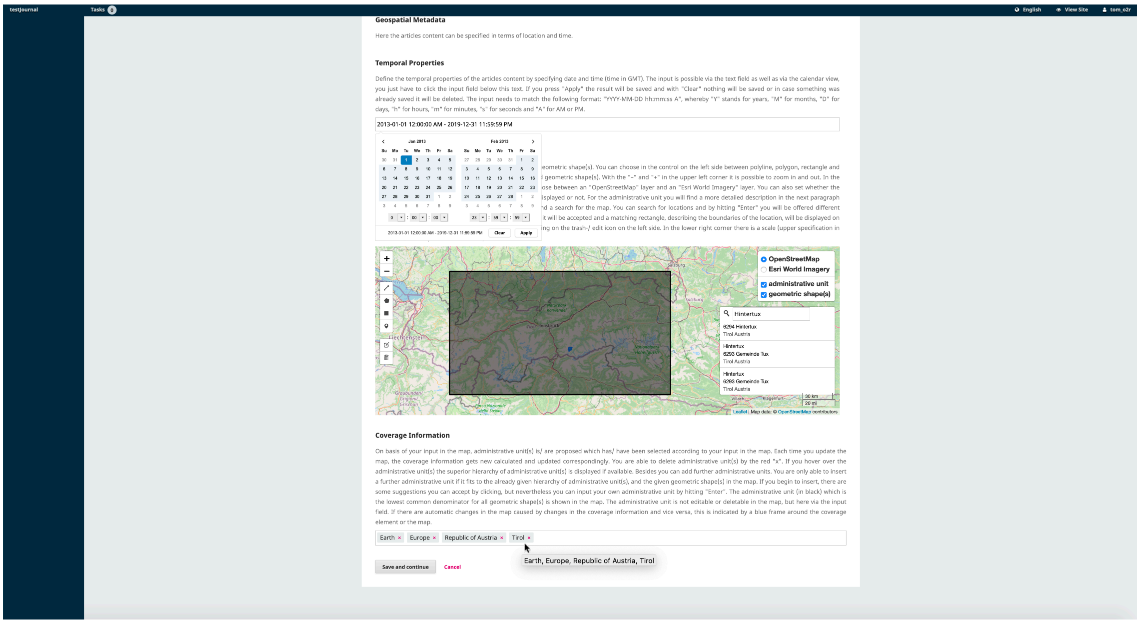

Figure 4.1: Screenshot of *geoOJS* – input of geospatial metadata (more focused screenshot available in the appendix in section C.1)

As described in chapter 3.2, previous work applied text recognition or pattern matching algorithms to derive geospatial metadata from scientific articles. But such fully automated workflows are not without shortcomings. Instead, we decided to streamline the user interaction to create geospatial metadata so that the user's understanding of the most suitable connections with location(s), area(s) and temporal properties is unambiguously recorded.

To define the temporal properties, the user can either define them in a textual input field by entering a correctly formatted string, or by clicking on that input field to open an interactive widget. In this widget the author can enter start and end to the second in a calendar view (see **/US18/**, **/US19/**). The temporal properties can be edited or deleted any time (see **/US23/**). To avoid inaccuracies, the time zone should be entered in GMT[62] (UTC±0) by the author.

Authors can either search for a location and accept the suggested bounding box or manually create one or more suitable geometric shape(s) on a map (see **/US12/**, **/US13/**, **/US14/**). If the author is searching for a location, suggestions are made based on the textual input (see **/US26/**). The author can accept a suggestion for which the appropriate standardized bounding box is displayed on the map, if available (see **/US28/**). Just as well the author can manually create one or more geometric shape(s) (see **/US12/**, **/US13/**). There is a choice for drawable geometric shapes between polyline, polygon, rectangle or marker (see **/US12/**). The drawn geometric shape(s), no matter created by author or by suggestion, can be edited or deleted (see **/US23/**, **/US29/**). The map can be controlled by pan and zoom and a scale is also shown (see **/US17/**). There is the choice between two different layers (see **/US15/**): either a satellite image or a topographic map. Furthermore, the author can use the layer control to decide whether the geometric shape(s) should be displayed.

If authors enter geometric shapes either manually or by search, a gazetteer is used to suggest a matching administrative unit's name to the author. Each time the geometric shape(s) on map change, the administrative units that fit for all geometric shapes displayed on the map are newly calculated and proposed (see **/US27/**).

An example shall clarify the procedure:

- There are currently two bounding boxes shown as geometric shapes on the map.
  - A bounding box for Berlin with the administrative units: `Earth, Europe, Federal Republic of Germany, Land Berlin, Berlin Stadt, Berlin`.
  - A bounding box for München with the administrative units: `Earth, Europe, Federal Republic of Germany, Bavaria, Upper Bavaria, Kreisfreie Stadt Muenchen, Muenchen Landeshauptstadt, Munich`.
- The administrative units that match both geometric shapes are suggested to the user: `Earth, Europe, Federal Republic of Germany`.

The administrative units are proposed in form of tags as coverage information. The author has the possibility to delete tags which should not be saved (see **/US29/**). By hovering over the tags the author can determine which administrative units are superior to each administrative unit represented by a tag. This allows the author to check the superior administrative units and to classify the tag correctly even if superior tags got deleted. Moreover, the lowest matching administrative unit for all geometric shape(s), i.e. the lowest common denominator, is displayed as a bounding box on the map (see **/US28/**). Only the lowest level is displayed to prevent an overload. In addition, the author can add further administrative units as long as they fit the given structure of administrative units and the currently displayed geometric shape(s) (see **/US11/**). It

[62] `https://en.wikipedia.org/wiki/Greenwich_Mean_Time` (Retr. 2020-10-07)

is not possible to add an administrative unit such as `Berlin`, if `Republic of France` is already stored as administrative unit. In the same way, it is not possible to add an administrative unit `Berlin` if a geometric shape for `Munich` is shown on the map. If the author tries to do so, there is a hint to either change the structure of already existing administrative units or the extent of the geometric shape(s), so that the administrative unit to be added is valid again. Nevertheless the author can add his own individual spatial entities. Of course these are no longer standardized, but we did not want to deny the authors possibility to store an individual spatial entity, because otherwise valuable information could be wasted.

All entries concerning the geospatial properties are not required, i.e. the authors only have to enter as much as they want.

The coverage information is initially offered in OJS for storage as an optional field. It represents the Dublin Core Coverage Element[63]. In previous versions of OJS the coverage element consisted of three different components. One field for geospatial or geographical terms, one for chronological periods and one for research sample characteristics which refer to the content of the scientific article. Since OSJ 2.4.x the coverage element in OJS is only a single optional field in which all the previously mentioned information can be stored. According the developers there were several reasons for this reduction. First, many of the metadata fields were bewildering, second the number of fields was overwhelming, third the fields were prone to getting bad data and fourth the stakes for submitters were very high.[64] The reasons are mainly plausible, but the solution of only one single coverage field should be reconsidered. As described in section 3.1, geospatial data are important for almost every scientific article, but nevertheless it is reasonable to offer their use as metadata optionally. Thus, for each journal can be decided whether coverage information as well as other metadata are of interest or not. This way only the relevant information are stored and an overload is avoided. However, if there is interest in storing geospatial metadata, a single field as coverage information offers only limited capacity. Certainly geospatial metadata can be specified like this, but one can not expect complete and standardized geospatial metadata. Especially the problem of the error frequency with a single field rather increases than decreases, since yet more different data have to be entered collected in a single field. Therefore we decided to specify the corresponding Coverage Element in OJS by standardized administrative units.

We decided to propose administrative units as spatial entities, because they are generally valid for all domains. The chances are higher that in a group of researchers from different fields, administrative units are more common to all than other spatial entities such as World Protected Areas, which only individual researchers understand.

Since available gazetteers for administrative units only provide results based on single pairs of coordinates and not on bounding boxes or geometric shapes with more than one pair of coordinates, a method is needed to provide the administrative unit for the whole geometric shape. For each pair of coordinates contained as edge in the geometric shape, a gazetteer is queried for the corresponding administrative units. Then, for each geometric shape, the administrative units matching for all coordinate pairs are recorded. If there are several geometric shapes on the map all administrative units that fit to all geometric shapes and their coordinates are stored. This way the administrative units are determined which are valid for all geometric shapes on the map.

[63] `https://www.dublincore.org/specifications/dublin-core/dcmi-terms/terms/coverage/` (Retr. 2020-10-07)
[64] Github Issue concerning additional metadata:
`https://github.com/pkp/pkp-lib/issues/1143#issuecomment-185168075` (Retr. 2020-10-07)

As described previously, the tags for the coverage element adjust each time the geometric shape(s) in the map change. Likewise, the administrative unit shown in the map adapts as soon as the user edits the tags for the coverage element. To make the user aware of the automatic changes, a blue frame is placed around the map and the coverage information if there are changes (see **/US31/**).

The geospatial metadata are stored in the database already used by OJS (see **/US22/**, **/US24/**, **/US34/**, **/US35/**, **/US36/**, **/US38/**). Coverage information and the temporal properties are stored in different fields.

The following data are stored for the coverage information:

`name`	The name of the administrative unit, which is also shown as tag, as `String`
`geonameId`	The geonameId, an unique id given by a gazetteer for each place, as `Integer`
`bbox`	The bbox, a bounding box got by the gazetteer separated in east, south, north and west, as `Floats`
`administrativeUnitSuborder`	The superior administrative units stored as array filled by `Strings`
`provenance`	The provenance, description as `String` and id as `Integer` to describe who created the administrative units and where the data come from

The following data are stored for the temporal properties:

`unixDateRange`	The start and end as unix timestamp in milliseconds in GMT stored as array filled by `Integers`
`provenance`	The provenance, description as `String` and id as `Integer` to describe who created the administrative units and where the data come from

In a third field a GeoJSON is stored, in which temporal and spatial properties as well as the coverage element are stored. A specification[65] of how this GeoJSON is structured can be found in the *geoOJS* wiki[66]. In the appendix in section A concrete database records for each type are shown as example.

4.1.2 Geospatial Metadata for Discovery of Scientific Articles in OJS

In OJS there is a web page for each scientific article, the so-called article view, where the scientific article is displayed with additional metadata. Besides to the usual details such as title, subtitle, abstract and publication date, the geospatial metadata are also displayed (see Figure 4.2).

The geospatial metadata are displayed in the article view below the abstract and thus contribute to the reader's understanding of the content of the scientific article (see **/US1/**, **/US2/**, **/US5/**). For the temporal properties the start and end are displayed in GMT to the second. A map displays the spatial properties of the scientific article's content (see **/US33/**). It has the same functions

[65] `https://github.com/tnier01/geoOJS/wiki/geoJSON-Specification` (Retr. 2020-10-07)
[66] `https://github.com/tnier01/geoOJS/wiki/` (Retr. 2020-10-07)

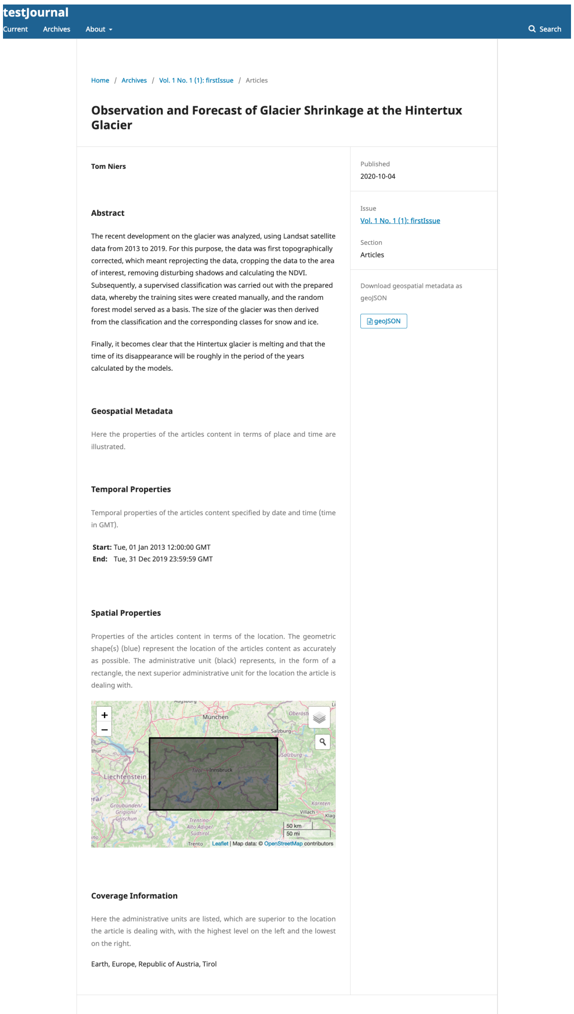

Figure 4.2: Screenshot of *geoOJS* – geospatial properties in the OJS article view (more focused screenshot available in the appendix in section C.2)

as the map in the submission process (see **/US15/**, **/US17/**) with some crucial differences: the reader can not add, edit or delete geometric shape(s) and the search does not automatically create a bounding box (see **/US14/**), it is a common search in which the focus of the map changes to the search entry if one of the search suggestions is accepted.

In addition to the lowest matching administrative unit for all geometric shape(s) displayed on the map, the coverage information is also displayed in text form. Descending from the left, the administrative units are listed (see **/US32/**, **/US46/**).

The GeoJSON with all geospatial metadata gathered is also available for the direct download in the right column of the article view (see **/US37/**, **/US45/**). Besides the geospatial metadata are available in the HTML header of the article page (see Figure 4.3, **/US44/**, **/US45/**). The name of the lowest matching administrative unit is stored as geotag, the name and the corresponding bounding box in the form of the DCMI box and the bounding box ISO 19139 compliant. Furthermore, the temporal properties are also stored as a meta element in the header ISO 8601 compatible. In the appendix in section B complete metadata elements are shown as an example.

```html
<meta name="geo.placename" content="Tirol">
<meta name="DC.box" content="name=Tirol; northlimit=47.7436172019182; southlimit=46.65180…80887848844; eastlimit=12.9666399491981; projection=EPSG3857">
<meta name="ISO 19139" content="<gmd:EX_GeographicBoundingBox><gmd:westBoundLongitude><gco:D…mal></gmd:northBoundLatitude></gmd:EX_GeographicBoundingBox>">
<meta name="ISO 8601" content="2013-01-01T12:00:00.000Z/2019-12-31T23:59:59.000Z">
```

Figure 4.3: HTML source code generated by *geoOJS* – geospatial metadata in the HTML document header of the article view

4.2 The *geoOJS* Plugin

The methodical implementation described above was developed in form of a plugin for OJS[67]. Initially, there was the consideration whether it is necessary to change the main code of OJS hard coded to enable the corresponding features of *geoOJS*. But OJS offers options for plugin integration[68], so it was decided to develop a plugin rather than change OJS itself. Particularly a plugin permits a future-safe development as the direct extending of the main code.

The plugin was created using an instance of OJS version 3.2.1.0. A version of the *geoOJS* plugin is available by the following link.

DOI 10.5281/zenodo.4079328

In this section we will explain the structure of the plugin, the concrete implementation and the dependencies used.

4.2.1 Architecture of the *geoOJS* Plugin

The plugin *geoOJS* was created according to the typical OJS plugin structure (see Figure 4.4). As implementation base a plugin template[69] provided by OJS was used. *geoOJS* is a generic plugin, because it deeply interferes with OJS.[70] Accordingly it is located in OJS in the folder structure of the generic plugins. We briefly describe the contained files and their function.

In the `geoOJSPlugin.inc.php` is a class named `geoOJSPlugin` which extends the `Generic Plugin` class. The `geoOJSPlugin` class is one of the core elements of the plugin. This class and the corresponding functions define name and description of the plugin, load dependencies, register hooks[71], extend the OJS templates with individual templates, and access the database.

The file `geoOJSPluginSettingsForm.inc.php` and the contained class `geoOJSPluginSettings Form` allow settings for the plugin. A corresponding template (`settings.tpl`) is loaded and applied.

All templates created individually for the plugin are contained in the folder `templates` in a corresponding substructure. They are stored in the same suborder as the initial templates given by OJS. The individual templates extend the templates initially provided by OJS.

Within the folder `js` are the JavaScript files that are necessary for the collection and representation of the geospatial metadata.

In addition, there are further files which are necessary for the implementation of *geoOJS*. The folder `locale` provides different languages, in this case only American English. All dependencies used by *geoOJS* are stored in the folder `enable_cdn_Off`. If the user does not want to actively retrieve the dependencies via links during use, they can also be loaded once before use. To do

[67]Github Repoository PKP OJS: `https://github.com/pkp/ojs` (Retr. 2020-10-04)
[68]Plugin guide OJS: `https://docs.pkp.sfu.ca/dev/plugin-guide/en/` (Retr. 2020-10-04)
[69]`https://github.com/pkp/pluginTemplate` (Retr. 2020-10-05)
[70]`https://docs.pkp.sfu.ca/dev/plugin-guide/en/categories#generic` (Retr. 2020-10-04)
[71]`https://docs.pkp.sfu.ca/dev/documentation/en/utilities-hooks` (Retr. 2020-10-05)

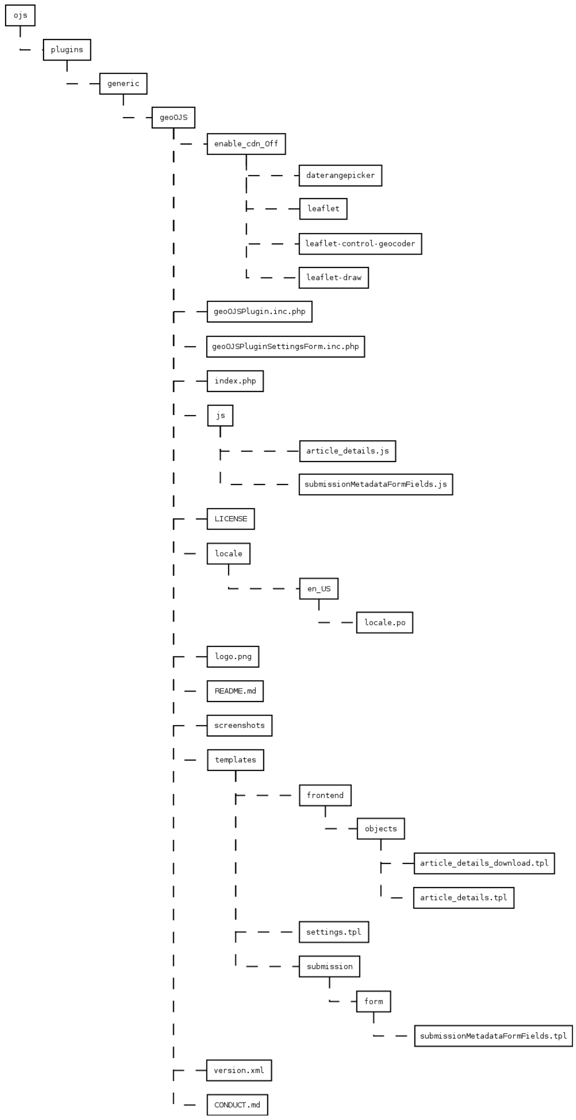

Figure 4.4: The *geoOJS* plugin

this the user has to adjust the CDN[72] setting in OJS's `config.inc.php` accordingly and add the dependencies in the `enable_cdn_Off` folder. The `version.xml` provides information required to load the plugin and the `index.php` file is required to load the correct plugin class. Other included files are the `LICENSE`, a `README.md`, a `CONDUCT.md`, a `logo.png` and other `screenshots` for the README.

4.2.2 Implementation of the *geoOJS* Plugin

In general the extension of a generic plugin in OJS is done by the `geoOJSPlugin` class in the `geoOJSPlugin.inc.php`. First, appropriate hooks are registered, which serve as entry points into the main code provided by OJS. Second, dependencies are loaded and JavaScript and CSS files are added accordingly. Third, functions are defined, to specify names and properties of the plugin,

[72] `https://en.wikipedia.org/wiki/Content_delivery_network` (Retr. 2020-10-04)

but mainly to call hooks and thus actively modify OJS. By means of the functions, existing templates can then be modified and extended by individual templates, and the structure of the database can be changed. The templates are `.tpl`-files[73], in which additional HTML elements can be stored, JavaScript files called and template variables retrieved. Figure 4.5 shows simplified how the `geoOJSPlugin` class integrates individual templates, using hooks. For the *geoOJS* plugin several different hooks were used, the effects of them are explained in the following.

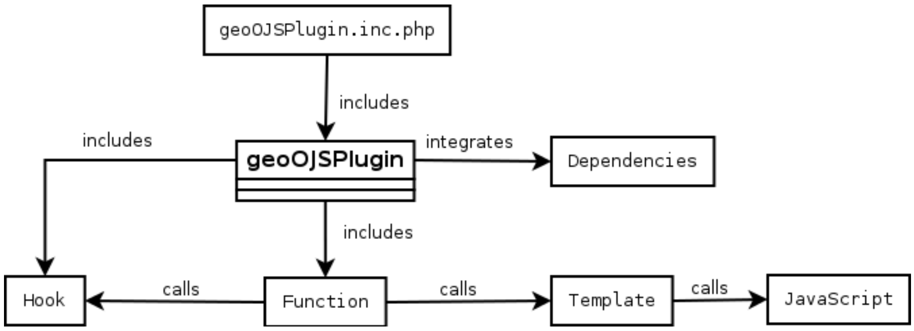

Figure 4.5: Simplified illustration of the interaction between the different parts of the *geoOJS* plugin

First, the hook `Templates::Submission::SubmissionMetadataForm::AdditionalMetadata` is used to extend the `submissionMetadataFormFields.tpl` template by an own individual *geoOJS* template, which has the same name due to the specifications of OJS. The step "3rd Enter Metadata" during the submission process is extended by an individual template and a corresponding JavaScript file (`submissionMetadataFormFields.js`). This enables the input of geospatial metadata by the author.

Second, the hooks `Templates::Article::Main` and `Templates::Article::Details` are used to extend the OJS article view. Therefore the two templates `article_details.tpl` and `article_details_download.tpl` are loaded to display the geospatial metadata in the article view and in the HTML header and to provide them as GeoJSON for download. For this purpose the JavaScript file `article_details.js` is included.

Third, the hook `Schema::get::publication` is used to extend the database. As mentioned in chapter 4.1.1, the geospatial metadata are stored in three fields of the OJS database. In our case it is stored in the OJS database which runs as a PostgreSQL database. There exists already a field for the administrative units, called `coverage`. For the spatial properties and the temporal properties there are two further fields created by the hook, called `geoOJS::temporalProperties` and `geoOJS::spatialProperties` (see Figure 4.6). Various types are available for the storage in the OJS database. This includes `Boolean`, `Integer`, `Number`, `String`, `Array` and `Object`.[74] However, we were not able to save the geospatial metadata as array or object. Therefore we decided to store the temporal properties as stringified array with two `integers`, the coverage information

[73] HTML/ Smarty Templates in OJS: `https://docs.pkp.sfu.ca/pkp-theming-guide/en/html-smarty.html` (Retr. 2020-10-05)

[74] Available datatypes for fields in OJS database: `https://github.com/pkp/pkp-lib/blob/222a310adcad88e335ec2d6990d9ff2623d3a992/classes/services/PKPSchemaService.inc.php#L237` (Retr. 2020-10-05)

as stringified array with JSON[75] objects and the spatial properties as stringified GeoJSON object. So the type `String` was used and the data were stringified or parsed accordingly. [76]

publication_id	locale	setting_name	setting_value
168		geoOJS::spatialProperties	{"type":"FeatureCollection","features":[{"type":"Feature","properties":{"provenance":"geometric shape created by user (edited the suggestion of the leaflet-control-geocoder by drawing)"},"geometry":{"type":"Polygon","coordinates":[[[11.666...
168	en_US	title	Observation and Forecast of Glacier Shrinkage at the Hintertux Glacier
168	en_US	type	
168		geoOJS::temporalProperties	[1357041600000,1577836799000]
168		categoryIds	a:0:{}
168	0	coverage	[{"name":"Earth","geonameId":6295630,"bbox":"not available","administrativeUnitSuborder": ["Earth"],"provenance":"administrative unit created by user (acceppting the suggestion of the geonames API , which was cr...
168	en_US	prefix	
168	en_US	rights	
168	en_US	source	
168	en_US	subtitle	
168	en_US	abstract	<div class="page" title="Page 1"> <div class="layoutArea">...
168	en_US	copyrightHolder	testJournal
168		copyrightYear	2020
168		issueId	1

Figure 4.6: Screenshot of the OJS DB by Postico[78]: Fields stored as metadata in the table `publication_settings` in the OJS database, geospatial metadata created by *geoOJS* highlighted in red

Fourth, the hook `Publication::edit` is used to store data in the OJS database. This hook is called each time the user wants to save changes to the metadata. The geospatial metadata are then stored in the individual fields of the database. Thereby the format described in chapter 4.1.1 is used. The hook `Publication::edit` is not only called in step "3rd Enter Metadata" during the submission process but also shortly before "Schedule for Publication", after the review process is finished. At this point the metadata can be changed again. Due to the scope of this thesis a modification of the geospatial metadata at this point was not implemented. However, there are already approaches how to implement an editing at this point as well.[79]

The geospatial metadata are recorded by JavaScript functions. In order to store them in the database with the hook `Publication::edit` they have to be passed to the PHP file. For this purpose the geospatial metadata are saved in hidden forms in the template by the JavaScript. Once the page is submitted via a POST request, OJS handles it and calls the `Publication::edit` hook, where the metadata are available via the $_POST[80] variable, from where it can be saved into the database (see Figure 4.7).[81]

Likewise, data already stored in the database must be displayed. First, if the user wants to edit his stored data again during the submission process and second, in the article view. In addition

[75]`https://www.json.org/json-en.html` (Retr. 2020-10-05)
[76]JSON.stringify() (`https://developer.mozilla.org/de/docs/Web/JavaScript/Reference/Global_Objects/JSON/stringify`) and JSON.parse() (`https://developer.mozilla.org/de/docs/Web/JavaScript/Reference/Global_Objects/JSON/parse`) (Retr. 2020-10-05)
[79]PKP community forum post: Approach to enable the editing of geospatial metadata in OJS right before "Schedule For Publication": `https://forum.pkp.sfu.ca/t/insert-in-submission-settings-table/61291/19?u=tnier01` (Retr. 2020-10-05)
[80]`https://www.php.net/manual/de/reserved.variables.post.php` (Retr. 2020-10-05)
[81]PKP community forum post: How to use a JS variable in php: `https://forum.pkp.sfu.ca/t/using-js-variable-in-php/62179/2` (Retr. 2020-10-05)

to the template, the above mentioned hooks pass template variables[82] in which the geospatial metadata are stored. The data are loaded from the database, passed on as template variables, retrieved by the JavaScript as value of hidden HTML elements, then prepared and displayed accordingly (see Figure 4.7).

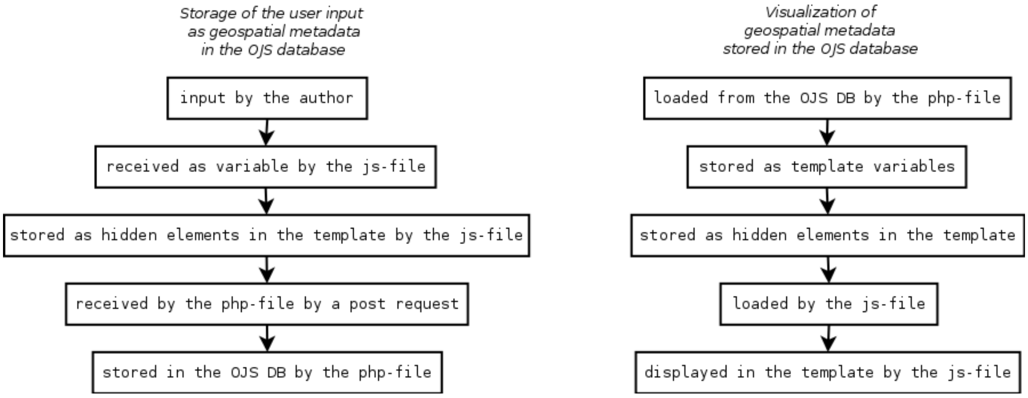

Figure 4.7: Chain how user input is stored as geospatial metadata in the OJS database (left) and how it is achieved from the OJS database to be visualized (right) by the *geoOJS* plugin

4.2.3 Dependencies Used for the *geoOJS* Plugin

For the implementation of *geoOJS* an external gazetteer and several dependencies were used.

A gazetteer is needed to suggest the administrative units to the user. Therefore GeoNames WebServices[83] was selected which offers a wide range of possible requests, whereby the following were used:

`http://api.geonames.org/searchJSON?`[84]	Used as autosuggest when the user wants to enter an administrative unit himself. Suitable suggestions for autocompletion are displayed. If the user accepts a suggestion, the ascii name, the geonameId and the bounding box (if available) will be saved from the response received from GeoNames.
`http://api.geonames.org/hierarchyJSON?`[85]	For each coordinate (latitude and longitude) transmitted to GeoNames, the corresponding hierarchy of administrative units is returned. In this way the administrative units that fit all geometric shapes can be calculated. In addition, instead of the coordinates, the geonameId is also queried to obtain the hierarchy of administrative units for the administrative units entered by the user.
`http://api.geonames.org/getJSON?`[86]	For a geonameId transmitted to GeoNames, the corresponding bounding box is returned. This allows to display the lowest administrative unit that fits all geometric shape(s).

[82] `https://docs.pkp.sfu.ca/pkp-theming-guide/en/template-variables.html` (Retr. 2020-10-05)
[83] `https://www.geonames.org/export/ws-overview.html` (Retr. 2020-10-05)

To implement the requirements the API requests mentioned above are necessary. GeoNames Webservice was the best choice to fulfill the requirements defined above. However, registration is required to use GeoNames Webservice. This can be provided by the administrator of the OJS instance. The file `geoOJSPluginSettingsForm.inc.php` with the class `geoOJSPluginSettingsForm` allows to define settings for the plugin. Accordingly, the user of *geoOJS* must first enter the corresponding username for GeoNames Webservice to be able to use the full scope of the plugin (see Figure 4.8). One has to keep in mind that there are certainly gazetteers with less features, but they do not require registration.[87]

Besides the input of the user name, other settings would also be conceivable. For example, one could specify which types of geometric shapes are available to the author, or whether a bounding box is automatically created during a map search or not.

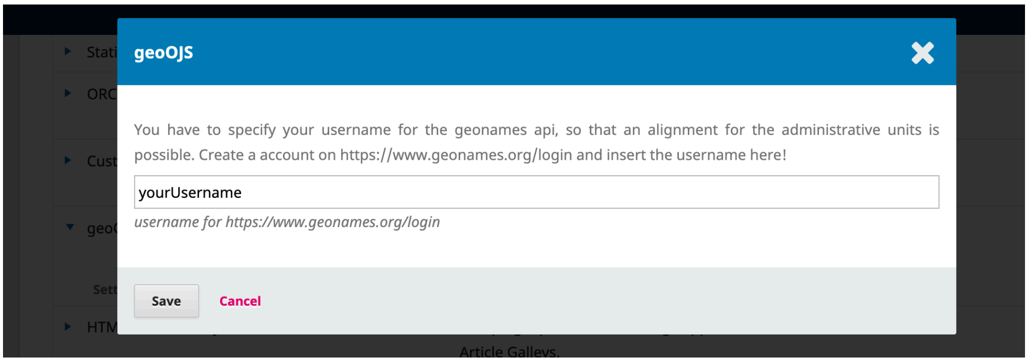

Figure 4.8: Screenshot of *geoOJS* – plugin settings to set usernames for GeoNames Webservice

For defining the temporal properties an interactive widget was included, the Date Range Picker[88]. Thus, the start and end time of the temporal properties can be defined exactly to the second.

A map was included so that the user can define and view the geospatial properties of the article. Therefore Leaflet[89], an open source JavaScript library for interactive maps, was used. To enable drawing geometric shapes on the map, Leaflet Draw[90] was integrated. Leaflet Draw allows drawing various geometric shapes, editing and deleting them. The Leaflet Control Geocoder[91] was also added to the map. A geocoder enables a textual search for places on the map.

To ensure that the coverage information and thus the administrative units can be changed interactively, they were implemented as tags.[92] OJS already uses tags for keywords regarding a scientific article, which can also be stored as metadata. The administrative units as tags are either added automatically after a corresponding map input or manually by the user. Moreover, the user can also delete them interactively.

[84] Documentation GeoNames search: `https://www.geonames.org/export/geonames-search.html` (Retr. 2020-10-05)
[85] Documentation GeoNames hierarchy: `https://www.geonames.org/export/place-hierarchy.html#hierarchy` (Retr. 2020-10-05)
[86] Documentation GeoNames get: `https://www.geonames.org/export/web-services.html#get` (Retr. 2020-10-05)
[87] Example for gazetteer without registration – photon: `http://photon.komoot.de/` (Retr. 2020-10-05)
[88] `https://github.com/dangrossman/daterangepicker` (Retr. 2020-10-05)
[89] `https://leafletjs.com/index.html` (Retr. 2020-10-05)
[90] `https://leaflet.github.io/Leaflet.draw/docs/leaflet-draw-latest.html` (Retr. 2020-10-05)
[91] `https://github.com/perliedman/leaflet-control-geocoder` (Retr. 2020-10-05)
[92] Tag-it: `https://github.com/aehlke/tag-it` (Retr. 2020-10-05)

The Date Range Picker and the tags need jQuery[93] to work. However, jQeury is already loaded initial by OJS and does not need to be additionally included by *geoOJS*.[94]

4.3 Evaluation of *geoOJS* Concerning the Requirements

geoOJS is an attempt to achieve well-defined geospatial metadata and to use it for discovery in OJS. With the implementation of geoOJS all but two of the requirements regarding the potential of integrating geospatial metadata in scholarly publishing were met. Thus *geoOJS* offers the possibility to store geospatial metadata for the discovery of scientific articles. Due to the scope of this thesis **/US3/** (a geospatial search filter for scientific articles) and **/US4/** (a map with all study areas of a journal) were not implemented. Nevertheless, with the storage of well-defined geospatial metadata provided by *geoOJS*, the basis for the implementation of these two user stories is set. For a combined representation of all study locations of a journal in a map, the geospatial metadata are available in the OJS database and must just be displayed on the corresponding web page. The geospatial metadata are stored in the HTML header of the article view in a standardized way and can thus be retrieved for external searches. Furthermore, the downloadable GeoJSON file can also be used by external services. Potential future work would be to provide a corresponding API endpoint for direct retrieval (see **/US43/**). With the provision of well-defined geospatial metadata *geoOJS* offers the starting point to exploit the potential for the discovery of scientific articles. An example for the benefit of this potential was implemented by the representation of the geospatial metadata in the article view.

We decided to streamline the user interaction to create geospatial metadata. Accordingly, the requirements **/US6/**, **/US7/**, **/US8/**, **/US9/**, **/US10/** were not realized. The decision was deliberately taken to use a user input inside OJS, so that the geospatial metadata are defined as well as possible in the author's interest. Moreover, a user input was the most obvious way for the implementation of a prototype like *geoOJS*. Nevertheless one should not disregard an automated collection of geospatial metadata. As described in chapter 3.2.1, automatic extraction could be the solution to provide a wide range of geospatial metadata for scientific article's content and thus to increase the potential of using geospatial metadata in scholarly publishing.

Due to the scope of this thesis it was decided not to give the author the possibility to add an own layer to the map. Accordingly, the user story **/ US16/** was not implemented. To still offer a certain amount of choice for different purposes, there is a choice between two different layers, a satellite image or a topographic map.

Regarding the time zone for the temporal properties GMT was generally chosen. To avoid ambiguity, e.g. if the author's time zone was determined automatically[95] but the entry is then made in a different time zone, a fixed time zone was chosen. Therefore the user stories **/US20/**, **/US21/** were not realized.

Due to the scope of this thesis, the type of spatial entities is limited and the user can not decide which one to use or upload an own layer. As described in chapter 4.1.1, the administrative units

[93] `https://jquery.com/` (Retr. 2020-10-05)

[94] jQuery loaded by OJS: `https://github.com/pkp/pkp-lib/blob/a59191c79146cab4f77431c8990e90dbc2db38f6/classes/template/PKPTemplateManager.inc.php#L801` (Retr. 2020-10-05)

[95] JavaScript function to determine the users timezone: `https://developer.mozilla.org/de/docs/Web/JavaScript/Reference/Global_Objects/Date/getTimezoneOffset` (Retr. 2020-10-06)

were chosen as spatial entities, since they are most comparable and generally understood. This is why the user story **/US30/** was not implemented.

A validation was not further implemented due to the scope of this thesis. Therefore the user stories **/US39/**, **/US40/**, **/US41/**, **/US42/** were not fulfilled. Only a limited validation by the author himself is possible in *geoOJS* by checking the input again on the map. This does not lead to the conclusion that validation is not necessary. To make the data as reliable as possible, a validation as described in chapter 3.5 is strongly recommended.

Thus the requirements regarding the *Potential of Integrating Geospatial Metadata in Scholarly Publishing* (see section 3.7.1) and *Achieving Geospatial Metadata by an Author Input* (see section 3.7.2) were largely met. The requirements concerning *Achieving Geospatial Metadata by Analyzing the Scientific Article* (see section 3.7.2) and *Validation of Geospatial Metadata in Scholarly Publishing* (see section 3.7.5) were not met at all. All remaining requirements are fulfilled by the implementation of *geoOJS*. Nevertheless, one must consider the limitation that only requirements based on literature and own ideas were implemented.

The *geoOJS* plugin provides an approach how well-defined geospatial metadata can be defined and stored in a database inside a scholarly publishing system. A novel semi-automated approach for the creation of standardized spatial metadata is applied. The user enters the location through a textual search, or a drawn geometric shape, and an appropriate administrative unit is suggested. In addition to the precise storage of the location, a standardized version of the location is automatically stored without additional effort for the user. It is shown how standardized geospatial metadata can help to classify the research work in the local and global context and thus to make it more comparable in relation to other scientific articles. We demonstrate how geospatial metadata can be presented and shared, no matter if the user depends on screen reading devices or not. Besides, we show possibilities how geospatial metadata can also be passed in a machine-readable form, so that spread of use increases. However, *geoOJS* reveals only a fraction of the potential that exists behind the integration of geospatial metadata in scholarly publishing systems. But *geoOJS* offers the basis to exploit this broad potential. Especially with regard to the previous possibilities defining geospatial metadata in OJS (coverage element), *geoOJS* implements an alternative approach to collect well-defined geospatial metadata and use them for the discovery of scientific articles.

Future work includes research into usability and usefulness of geospatial metadata for the discovery of articles, a search engine across OJS instances, geospatial and temporal filters in article search, and validation of geospatial metadata as part of the review process. We plan to release *geoOJS* in the OJS plugin gallery so that the increasing number of independent and Open Access journals may benefit from it.

5 Conclusion

The acquisition of geospatial metadata for the content of scientific articles has a high potential and can improve the discovery in scholarly publishing. Especially the search for scientific articles can be improved because the new query parameter time and location are available. The most-requested prerequisite for this is that the data are stored in a standardized format but also standardized with regard to precision in order to ensure comparability between the articles and their geospatial metadata. To obtain data that are as accurate as possible, the most reliable method is an input by the author. Nevertheless, automatic approaches such as text recognition should not be disregarded, as they are a promising way to create a large amount of geospatial metadata, especially for articles that already exist. Validation of the geospatial metadata, whether automated or as part of the review process, is recommended to obtain credible data. Ultimately, the crucial factor is to make the well-defined geospatial metadata accessible in every respect, both for humans and machines. Only then geospatial metadata can exploit full potential for discovery in scholarly publishing.

geoOJS offers a novel way for authors to provide spatial properties of research works when submitting an article to a journal in OJS. In addition to precise spatial properties, standardized administrative units can be added to the article semi-automatically without additional effort for the author. With the provision of geospatial metadata both in the form of maps and text as well as in a machine readable form in the HTML source code of article's landing page in a semantically meaningful way, *geoOJS* shows how geospatial metadata can contribute to the discovery of scientific articles.

References

Elsevier. (n.d.). *Include interactive maps with your next article.* Retrieved 2020-06-09, from `https://www.journals.elsevier.com/global-environmental-change/news/interactive-maps`

Hendrickx, D., Dujardin, J.-C., Pickering, J., & Alvar, J. (2010). The leishmaniasis e-compendium: a geo-referenced bibliographic tool. *Trends in parasitology.-London*, *26*(11), 515–516. doi: 10.1016/j.pt.2010.06.015

Howell, R. G., Petersen, S. L., Balzotti, C. S., Rogers, P. C., Jackson, M. W., & Hedrich, A. E. (2019). Using webgis to develop a spatial bibliography for organizing, mapping, and disseminating research information: a case study of quaking aspen. *Rangelands*, *41*(6), 244–247. doi: 10.1016/j.rala.2019.10.001

Karl, J. W. (2019). Mining location information from life-and earth-sciences studies to facilitate knowledge discovery. *Journal of Librarianship and Information Science*, *51*(4), 1007–1021. doi: 10.1177/0961000618759413

Karl, J. W., Gillan, J. K., & Herrick, J. E. (2013). Geographic searching for ecological studies: a new frontier. *Trends in Ecology & Evolution*, *28*(7), 383–384. doi: 10.1016/j.tree.2013.05.001

Karl, J. W., Herrick, J. E., Unnasch, R. S., Gillan, J. K., Ellis, E. C., Lutters, W. G., & Martin, L. J. (2013). Discovering ecologically relevant knowledge from published studies through geosemantic searching. *BioScience*, *63*(8), 674–682. doi: 10.1525/bio.2013.63.8.10

Keßler, C., Maué, P., Heuer, J. T., & Bartoschek, T. (2009). Bottom-up gazetteers: Learning from the implicit semantics of geotags. In *International conference on geospatial sematics* (pp. 83–102).

Kmoch, A., Uuemaa, E., Klug, H., & Cameron, S. G. (2018). Enhancing location-related hydrogeological knowledge. *ISPRS International Journal of Geo-Information*, *7*(4), 132. doi: 10.3390/ijgi7040132

Konkol, M., & Kray, C. (2019). In-depth examination of spatiotemporal figures in open reproducible research. *Cartography and Geographic Information Science*, *46*(5), 412–427.

Magliocca, N. R., Rudel, T. K., Verburg, P. H., McConnell, W. J., Mertz, O., Gerstner, K., … Ellis, E. C. (2015). Synthesis in land change science: methodological patterns, challenges, and guidelines. *Regional environmental change*, *15*(2), 211–226.

Margulies, J. D., Magliocca, N. R., Schmill, M. D., & Ellis, E. C. (2016). Ambiguous geographies: connecting case study knowledge with global change science. *Annals of the American Association of Geographers*, *106*(3), 572–596. doi: 10.1080/24694452.2016.1142857

Martin, L. J., Blossey, B., & Ellis, E. (2012). Mapping where ecologists work: biases in the global distribution of terrestrial ecological observations. *Frontiers in Ecology and the Environment*, *10*(4), 195–201. doi: 10.1890/110154

Mohr, M. (2017). *Machine-based extraction of location information from digital publications.* (MSc thesis, Institute for Geoinformatics, University of Münster)

Nüst, D., & Qamaz, Y. (2020). *geoextent.* Zenodo. Retrieved from `https://o2r.info/geoextent/` doi: 10.5281/zenodo.3925693

A place for everything. (2008). *Nature*, *453*(7191), 2–2. Retrieved from `https://doi.org/10.1038/453002a` doi: 10.1038/453002a

Scheider, S., Degbelo, A., Kuhn, W., & Przibytzin, H. (2014). Content and context description—how linked spatio-temporal data enables novel information services for libraries. *GIS Sci*, *4*, 138–149.

Schirmacher, H. (n.d.). „gute “software entwickeln–auch im kleinen team. Retrieved 2020-09-26, from https://edoc.bbaw.de/opus4-bbaw/files/3075/BBAW_DH_Kolloquium_20180601_Schirmacher.pdf

Shapiro, J. T., & Báldi, A. (2012). Lost locations and the (ir) repeatability of ecological studies. *Frontiers in Ecology and the Environment*, *10*(5), 235–236. doi: 10.1890/12.WB.015

Young, A. L., & Lutters, W. G. (2017). Infrastructuring for cross-disciplinary synthetic science: Meta-study research in land system science. *Computer Supported Cooperative Work (CSCW)*, *26*(1-2), 165–203. doi: 10.1007/s10606-017-9267-z

Appendix

A Examples for Storage of Geospatial Metadata in the OJS Database

A.1 Example for the Temporal Properties in the OJS Database

This is an example for the storage of temporal properties in the OJS database in the field `geoOJS::temporalProperties`:


```
[1357041600000,1577836799000]
```


A.2 Example for the Spatial Properties in the OJS Database

This is an example for the storage of spatial properties in the OJS database in the field `geoOJS::spatialProperties`:


```
{
    "type": "FeatureCollection",
    "features": [
        {
            "type": "Feature",
            "properties": {
                "provenance": {
                    "description": "geometric shape created by user (drawing)",
                    "id": 11
                }
            },
            "geometry": {
                "type": "Polygon",
                "coordinates": [
                    [
                        [
                            11.65254592895508,
                            47.04149820647921
                        ],
                        [
                            11.657180786132814,
                            47.04570938567778
                        ],
                        [
                            11.663532257080078,
                            47.04980326897738
                        ],
                        [
                            11.663703918457031,
                            47.051908572325715
                        ],
                        [
                            11.667137145996096,
                            47.05296119283581
                        ],
                        [
                            11.671600341796875,
                            47.04910148272694
                        ],
                        [
                            11.678981781005861,
                            47.04968630524354
                        ],
                        [
                            11.676578521728516,
                            47.05635282870732
                        ],
                        [
                            11.681213378906252,
                            47.05705451954216
                        ],
                        [
                            11.682071685791016,
                            47.06231690648144
                        ],
                        [
                            11.676921844482422,
                            47.06512330047693
                        ],
                        [
                            11.679496765136719,
                            47.06629258770576
```

```
                        ],
                        [
                            11.678123474121096,
                            47.06933261446594
                        ],
                        [
                            11.687736511230469,
                            47.06968337562739
                        ],
                        [
                            11.690654754638674,
                            47.07225555361204
                        ],
                        [
                            11.687049865722658,
                            47.074009240155284
                        ],
                        [
                            11.679153442382814,
                            47.07260629553734
                        ],
                        [
                            11.676406860351564,
                            47.07260629553734
                        ],
                        [
                            11.674003601074219,
                            47.075762868990545
                        ],
                        [
                            11.669197082519531,
                            47.07611358783266
                        ],
                        [
                            11.666622161865234,
                            47.07424306066756
                        ],
                        [
                            11.663532257080078,
                            47.076815018591994
                        ],
                        [
                            11.658897399902346,
                            47.07225555361204
                        ],
                        [
                            11.652717590332031,
                            47.07587977552774
                        ],
                        [
                            11.648597717285158,
                            47.07330777246297
                        ],
                        [
                            11.646366119384767,
                            47.06629258770576
                        ],
                        [
                            11.652202606201174,
                            47.06278464907349
                        ],
                        [
                            11.658897399902346,
                            47.06091365408244
                        ],
                        [
                            11.656494140625002,
                            47.05939342230417
                        ],
                        [
                            11.651687622070314,
                            47.06149834707057
                        ],
                        [
                            11.64379119873047,
                            47.06126447064478
                        ],
                        [
                            11.64653778076172,
                            47.05799009295849
                        ],
                        [
                            11.651344299316408,
                            47.05518332346321
                        ],
                        [
                            11.641387939453127,
```

```
                            47.051791613208785
                        ],
                        [
                            11.640357971191408,
                            47.046996068572405
                        ],
                        [
                            11.648597717285158,
                            47.04828272043085
                        ],
                        [
                            11.657352447509767,
                            47.051908572325715
                        ],
                        [
                            11.65529251098633,
                            47.04863362009673
                        ],
                        [
                            11.651000976562502,
                            47.04839968724263
                        ],
                        [
                            11.65254592895508,
                            47.04149820647921
                        ]
                    ]
                ]
            }
        }
    ],
    "administrativeUnits": [
        {
            "name": "Earth",
            "geonameId": 6295630,
            "bbox": "not available",
            "administrativeUnitSuborder": [
                "Earth"
            ],
            "provenance": {
                "description": "administrative unit created by user (acceppting the suggestion of the geonames API , which was created on basis of a geometric shape input)",
                "id": 23
            }
        },
        {
            "name": "Europe",
            "geonameId": 6255148,
            "bbox": {
                "east": 41.73303985595703,
                "south": 27.6377894797159,
                "north": 80.76416015625,
                "west": -24.532675386662543
            },
            "administrativeUnitSuborder": [
                "Earth",
                "Europe"
            ],
            "provenance": {
                "description": "administrative unit created by user (acceppting the suggestion of the geonames API , which was created on basis of a geometric shape input)",
                "id": 23
            }
        },
        {
            "name": "Republic of Austria",
            "geonameId": 2782113,
            "bbox": {
                "east": 17.1620685652599,
                "south": 46.3726520216244,
                "north": 49.0211627691393,
                "west": 9.53095237240833
            },
            "administrativeUnitSuborder": [
                "Earth",
                "Europe",
                "Republic of Austria"
            ],
            "provenance": {
                "description": "administrative unit created by user (acceppting the suggestion of the geonames API , which was created on basis of a geometric shape input)",
                "id": 23
            }
        },
        {
            "name": "Tirol",
            "geonameId": 2763586,
```

```json
            "bbox": {
                "east": 12.9666399491981,
                "south": 46.6518080714719,
                "north": 47.7436172019182,
                "west": 10.0980887848844
            },
            "administrativeUnitSuborder": [
                "Earth",
                "Europe",
                "Republic of Austria",
                "Tirol"
            ],
            "provenance": {
                "description": "administrative unit created by user (acceppting the suggestion of the geonames API , which was created on
                       basis of a geometric shape input)",
                "id": 23
            }
        }
    ],
    "temporalProperties": {
        "unixDateRange": "[1357041600000,1577836799000]",
        "provenance": {
            "description": "temporal properties created by user",
            "id": 31
        }
    }
}
```

A.3 Example for the Coverage Information in the OJS Database

This is an example for the storage of the coverage information in the OJS database in the field `coverage`:


```
[
    {
        "name": "Earth",
        "geonameId": 6295630,
        "bbox": "not available",
        "administrativeUnitSuborder": [
            "Earth"
        ],
        "provenance": {
            "description": "administrative unit created by user (acceppting the suggestion of the geonames API , which was created on basis of a geometric shape input)",
            "id": 23
        }
    },
    {
        "name": "Europe",
        "geonameId": 6255148,
        "bbox": {
            "east": 41.73303985595703,
            "south": 27.6377894797159,
            "north": 80.76416015625,
            "west": -24.532675386662543
        },
        "administrativeUnitSuborder": [
            "Earth",
            "Europe"
        ],
        "provenance": {
            "description": "administrative unit created by user (acceppting the suggestion of the geonames API , which was created on basis of a geometric shape input)",
            "id": 23
        }
    },
    {
        "name": "Republic of Austria",
        "geonameId": 2782113,
        "bbox": {
            "east": 17.1620685652599,
            "south": 46.3726520216244,
            "north": 49.0211627691393,
            "west": 9.53095237240833
        },
        "administrativeUnitSuborder": [
            "Earth",
            "Europe",
            "Republic of Austria"
        ],
        "provenance": {
            "description": "administrative unit created by user (acceppting the suggestion of the geonames API , which was created on basis of a geometric shape input)",
            "id": 23
        }
    },
    {
        "name": "Tirol",
        "geonameId": 2763586,
        "bbox": {
            "east": 12.9666399491981,
            "south": 46.6518080714719,
            "north": 47.7436172019182,
            "west": 10.0980887848844
        },
        "administrativeUnitSuborder": [
            "Earth",
            "Europe",
            "Republic of Austria",
            "Tirol"
        ],
        "provenance": {
            "description": "administrative unit created by user (acceppting the suggestion of the geonames API , which was created on basis of a geometric shape input)",
            "id": 23
        }
    }
]
```

B Examples for Standardized Storage of Geospatial Metadata in the HTML Document Header of the OJS Article View

B.1 Example for a Geo Meta Tag in the HTML Document Header of the OJS Article View

```
<meta name="geo.placename" content="Tirol">
```

B.2 Example for DCMI Box Encoding Scheme in the HTML Document Header of the OJS Article View

```
<meta name="DC.box" content="name=Tirol; northlimit=47.7436172019182; southlimit=46.6518080714719; westlimit=10.0980887848844; eastlimit=12.96
      66399491981; projection=EPSG3857"/>
```

B.3 Example for ISO 19139 in the HTML Document Header of the OJS Article View

```
<gmd:EX_GeographicBoundingBox><gmd:westBoundLongitude><gco:Decimal>10.0980887848844</gco:Decimal></gmd:westBoundLongitude><gmd:
      eastBoundLongitude><gco:Decimal>12.9666399491981</gco:Decimal></gmd:eastBoundLongitude><gmd:southBoundLatitude><gco:Decimal>46.65180807
      14719</gco:Decimal></gmd:southBoundLatitude><gmd:northBoundLatitude><gco:Decimal>47.7436172019182</gco:Decimal></gmd:northBoundLatitude
      ></gmd:EX_GeographicBoundingBox>
```

B.4 Example for ISO 8601 in the HTML Document Header of the OJS Article View

```
<meta name="ISO 8601" content="2013-01-01T12:00:00.000Z/2019-12-31T23:59:59.000Z">
```

C Screenshots of *geoOJS*

C.1 Screenshot of *geoOJS* – Input of Geospatial Metadata

C.2 Screenshot of *geoOJS* – Geospatial Properties in the OJS Article View

of interest, removing disturbing shadows and calculating the NDVI. Subsequently, a supervised classification was carried out with the prepared data, whereby the training sites were created manually, and the random forest model served as a basis. The size of the glacier was then derived from the classification and the corresponding classes for snow and ice.

Finally, it becomes clear that the Hintertux glacier is melting and that the time of its disappearance will be roughly in the period of the years calculated by the models.

Download geospatial metadata as geoJSON

geoJSON

Geospatial Metadata

Here the properties of the articles content in terms of place and time are illustrated.

Temporal Properties

Temporal properties of the articles content specified by date and time (time in GMT).

Start: Tue, 01 Jan 2013 12:00:00 GMT
End: Tue, 31 Dec 2019 23:59:59 GMT

Spatial Properties

Properties of the articles content in terms of the location. The geometric shape(s) (blue) represent the location of the articles content as accurately as possible. The administrative unit (black) represents, in the form of a rectangle, the next superior administrative unit for the location the article is dealing with.

Coverage Information

Here the administrative units are listed, which are superior to the location the article is dealing with, with the highest level on the left and the lowest on the right.

Earth, Europe, Republic of Austria, Tirol

Declaration of Academic Integrity

I hereby confirm that this thesis on *Geospatial Metadata for Discovery in Scholarly Publishing* is solely my own work and that I have used no sources or aids other than the ones stated. All passages in my thesis for which other sources, including electronic media, have been used, be it direct quotes or content references, have been acknowledged as such and the sources cited.

Münster, October 13, 2020 ______________________________

I agree to have my thesis checked in order to rule out potential similarities with other works and to have my thesis stored in a database for this purpose.

Münster, October 13, 2020 ______________________________