Matthias Lang et al.
http://dx.doi.org/10.15496/publikation-43218

Spacialist – A Virtual Research Environment for the Spatial Humanities

Matthias Lang
matthias.lang@
uni-tuebingen.de

Michael Derntl
Benjamin Glissmann
Vinzenz Rosenkranz
Dirk Seidensticker
David Kirschenheuter

Universität Tübingen,
eScience-Center

Abstract

Many archaeological research projects generate data and tools that are unusable or abandoned after the funding period ends. To counter this unsustainable practice, the Spacialist project was tasked to create a virtual research environment that offers an integrated, web-based user interface to record, browse, analyze, and visualize all spatial, graphical, textual and statistical data from archaeological or cultural heritage research projects. Spacialist is developed as an open-source software platform composed of modules providing the required functionality to end-users. It builds on controlled multi-language vocabularies and an abstract, extensible data model to facilitate data recording and analysis, as well as interoperability with other projects and infrastructures. Development of Spacialist is driven by an interdisciplinary team in collaboration with various pilot projects in different areas of archaeology. To support the complete research lifecycle, the platform is being integrated with the University's research-data archive, guaranteeing long-term availability of project data.

Keywords: spatial humanities, virtual research environments, digital archaeology

Introduction

Nearly every archaeological project uses some kind of digital data management system. That could be a complex folder and file structure, a database system or a geographical information system (GIS). In many cases, all these tools are combined in more or less complex environments requiring a significant amount of maintenance. Usually, development, implementation and maintenance are in the hands of archaeologists often with limited knowledge of these information technologies. In our experience, such systems tend to be unsustainable because they lack a thorough requirements analysis and cannot make adjustments. Consequently, these environments will be very specific for a single project, and the reuse in comparable projects will be largely impossible without significant, often complex adaptations. The lack of a thorough and complete documentation as well as the frequent change of the responsible staff makes matters even worse. Thus, most new archaeological projects will start from scratch with a new system setup, repeating the same mistakes and facing the same issues over and over.

Furthermore, in most cases, reuse of data is also not intended, and projects do not have a data management plan to guarantee long-term sustainability and reusability of their data sets and analyses. Facing the downsides of these practices, many funding agencies (e.g., the German Research Foundation) have made data management and data archiving and sustainability plans mandatory for new research project proposals. As we observe in our daily consulting practice at the eScience-Center at University of Tübingen, researchers in the humanities are struggling to create such plans because of a lack of knowledge about methods and tools that support this effort during and after the project. The complete loss of data after a short period of time following the project's end can be observed in many cases.

In the last decades, a variety of projects tried to develop a data management system, aiming to solve the aforementioned problems with a single tool. Powerful tools like ARK (Eve and Hunt 2008), Heurist (Johnson 2011), and iDAI.field (Henze et al. 2013) were developed, but never widely accepted in the archaeological community. These comprehensive tools might be very useful for large projects with powerful IT support. In most archaeological undertakings, however, these systems would overburden the staff with too many ways of managing the data and complicated installation and maintenance procedures.

In 2009, the ArchGate project started with the aim to develop a modular digital research environment for small projects with limited needs and limited IT support (Lang 2012). Like the projects mentioned before, ArchGate was also never widely accepted by the targeted audience. We underestimated the heterogeneity of the archaeological projects resulting in constant project-specific adjustments, which were neither readily transferable to other projects nor contributing to the system as a whole. The main cause of this problem was a lack of abstraction in the data model, which was not flexible enough to store certain types of information generated by the researchers. We ended up with a wide range of different versions of the same tool with diverging data models, tables and functionalities, deployed on different computers. Since ArchGate was designed as a locally deployed Java desktop application, updates and patches had to be applied to every single ArchGate instance. We also underestimated the workload necessary to maintain all these instances, resulting in delays and dissatisfied users, which caused even more project-specific adjustments. After ArchGate ran out of funding most of our users simply did not have the financial capability nor the willingness to pay for necessary changes and updates to the system. Therefore, we decided to give up the tool and relaunch it under the brand name "Spacialist" based on the lessons we learned from ArchGate.

For us it was clear that we needed a higher degree of abstraction of our data model to manage all information recorded by the researchers, without requiring project-specific adjustments to deployed instances. Furthermore, we assigned permanent staff maintaining and updating the system in constant contact with the end-users to avoid common project-funding issues inherent in short-term research projects. After the project funding ends, a sustainable business model will be essential to guarantee the end-users a long-term maintenance of the system and their data.

The Spacialist Approach

It is evident that most archaeological projects have a set of common requirements and constraints related to collaboration, recording and editing as well as visualization and analysis of research data, including georeferenced information. However, many projects—even within a specific subdomain of archaeology—maintain highly heterogeneous sets of project-specific requirements. This heterogeneity can, for instance, range from simple labeling and language issues to more complex issues like different conceptualizations and organization of objects and relationships within the projects' (sub-)domains. Also, the set of features required for offering tailored tools for particular projects, work practices within a project team, and supporting these over the life span of a project as well as afterwards can vary significantly. While one project, for instance, might organize data using a stratigraphic organization like a Harris matrix, other projects might rely on a hierarchical taxonomy or geographic map as primary organization and interaction metaphor. The following two subsections are intended to set out the approach dealing with this heterogeneity in Spacialist while avoiding reinventing the wheel. This relates to the Spacialist project setup (Section 2.1) on the one hand, and the principles of the Spacialist software (Section 2.2) on the other hand.

Project Setup

A prototype of Spacialist is being developed over a two-year period in a publicly funded project through the eScience initiative of the state government of Baden-Württemberg, Germany. One of the keys to success of the project and thus obtaining a high quality of the software is an early and tight involvement of pilot project partners, who help provide the development team with functional requirements, feature requests and feedback (Derntl et al. 2015). To ensure a smooth communication between collaborating researchers and the Spacialist software developers, the

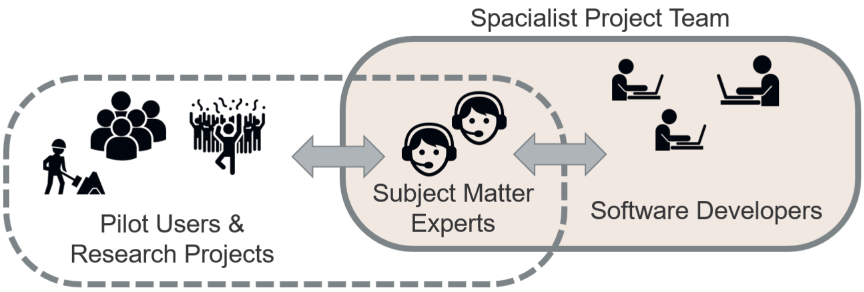

Figure 1. Project collaboration setup.

eScience-Center employs in addition two part-time subject matter experts – archaeologists with a strong computational background and field-work experience – as part of the project team to mediate and translate (see Figure 1). The whole project is coordinated by permanent staff.

To increase the visibility of Spacialist for research projects and to simplify participation for external developers, we rely on an open-source software strategy, which includes (a) the exclusive reliance on open-source libraries and frameworks, and (b) the commitment to publicly host a software repository on GitHub[1] under the MIT license[2], a strongly permissive open-source license. Moreover, most parts of the software process are openly communicated via GitHub milestones, releases, and issue tracking features. Any user may post issues, bugs and feature requests, and can get in touch with the developer team.

Technology Pillars

Learning from previous failures like the ones addressed in the introduction, we initially devised a small set of technological pillars that should later guide each and every technical decision in Spacialist. These pillars rest mainly on the principle of abstraction, which enables breaking down the complexity of a software system into levels that encapsulate certain system aspects and that build and rely upon each other (e.g., Vogel et al. 2011). The three pillars are:

- **Modular, extensible application platform**. In Spacialist a set of core modules can be combined and extended with add-on modules that are required or useful in a specific project. Spacialist is deployed as a hosted platform (either self-hosted or third-party hosted) on the web, and for each project a custom configuration of core and extension modules is instantiated to serve as the project-branded Spacialist application using a specific web address. The system architecture is described in Section 4.1 and the current ecosystem of modules is explained in Section 4.3.

- **Domain-independent, extensible data model**. The data model of the Spacialist core remains independent of any specific project or domain. It allows defining project-specific *context types* (representing any type of object, thing or entity in the project domain, e.g., a survey area or a find), their relationships, and the attributes that identify and describe individual instances of context types. The core data model can be extended by add-on modules. It is described in detail in Section 4.2.

- **Thesaurus-based domain representation**. Knowledge representation in all Spacialist instances is based on a simple, highly abstract representation of concepts and their relationships in the target domain of the projects. This allows hierarchical knowledge organization. This thesaurus-based approach is used both (a) for the labeling of user interface elements and (b) for the definition of project-specific context types, attributes, and value enumerations. The approach is described in detail in Section 4.4.

Application Cases

The development of Spacialist is driven and accompanied by a diverse range of research projects mainly from archaeology and cultural heritage. These

1 https://github.com/eScienceCenter

2 https://opensource.org/licenses/MIT

research projects have their own funding and have committed themselves to using Spacialist as a virtual research environment. This is a win-win situation in that it offers real-world application cases for the Spacialist team, while the participating projects benefit from tailored software modules and support. To get an impression of the kind of projects supported by Spacialist, we present two selected pilot projects in more detail.

Settlement and Society in Pre-Modern Oman

Large parts of the Sultanate of Oman are characterized by a mainly arid and hostile life environment. Less than one hundred years ago, before the modernization of the country in the second half of the twentieth century, sedentary live was mainly restricted to scattered oasis settlements of different shapes and sizes (Jones 2015). Oases formed microcosms consisting of various settlement and fortification buildings, infrastructure, date plantations and agricultural areas, as well as complex irrigation systems guaranteeing the survival in these environmental conditions. Archaeological excavations and remains of ancient monuments still visible on the surface point to settlement sites of the third millennium BC located close to modern oases, posing the questions of similarities or differences in site location preferences and adaptation to environmental pressure.

Such questions shall be faced by the recently initialized project "Settlement and Society in pre-modern Oman" – an interdisciplinary collaboration between the institutes of Islamic Studies, Near Eastern Archaeology and the eScience-Center of the University of Tübingen together with the Ministry of Heritage and Culture and the Ministry of Endowment and Religious Affairs of the Sultanate of Oman. The primary aims are (1) the examination and diachronic comparison between these two forms of subsistence separated by four millennia; (2) documentation of the abandoned mudbrick settlements by modern aerial and terrestrial scanning methods; and (3) conservation of knowledge about water and oasis management, which was passed on mainly orally over the generations.

These aims and the specific project setting come with several constraints and requirements, which guided the selection of Spacialist as the virtual research environment to support the entire project lifecycle. Some of the most important ones are listed below:

- A key characteristic of the Oman project is its multidisciplinary nature and the close collaboration with the local ministries and authorities allowing the combination of different methods and techniques from varying sciences. Such an approach may result in a broad and widespread understanding of the subject matter but also comes with some risks. Due to different methods, terminologies and data sets, the created information is often collected and stored by each party in their own databases and software environments, being only partly accessible by or portable to other partners, thus preventing direct and efficient scientific exchange. To avoid such problems, a collaborative tool for acquiring, storing and linking data based on standardized formats was required.

- A controlled vocabulary was required to allow translating labels for all used concepts and entities into different languages. This would facilitate the collaboration and exchange with local authorities in Oman as well as colleagues from around the world. The users can work in their familiar language without accidentally mixing up concepts.

- Since enormous amounts of various digital data are being produced and used within the research project, tools to upload, download, view and tag different data elements and sets like photos, scans, documents, geodata and 3D models were needed. This allows all partners to use these data without distributing them manually and using special viewers.

- The complexity of the involved sciences and the general subject matter makes it impossible to cover all possibilities in predefined data forms or controlled vocabularies. Therefore, an important requirement was to allow the creation and ingestion of new concepts over time, the reorganization and consolidation of the thesaurus, as well as the adaptation and definition of existing and new form sheets

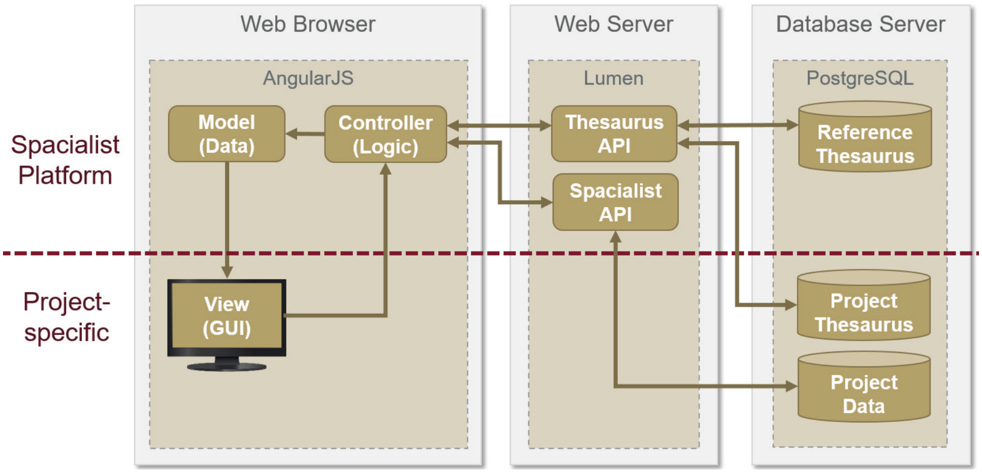

Figure 2. Spacialist layered system architecture.

to enter data, respectively, without requiring technical knowledge about operating a database system. This would make the project partners more agile and also facilitate later extensions and adaptations of the virtual research environment or the incorporation of new methods or approaches, for instance, in follow-up projects.

- The need to link data to a given concept (context or type) bundles all available information in one place. This allows internal researchers as well as external specialist to get much needed background information and to stay up to date.

Steinlachtal

The valleys of the Steinlach and Wiesaz Rivers, located in Southwestern Germany at the foot of the Swabian Jura, between the cities Tübingen and Hechingen, have a rich archaeological record. By using landscape-archaeological approaches one can try to reconstruct the ancient settlement structure and extract information from the record about how people have used and experienced the landscape. Due to its location at the foot of the Swabian Jura, the Steinlach river valley represents a transitional zone. First, a communication path alongside the Swabian Jura directs right through the valley. Second, the Wiesaz valley represents one of several valleys, which represent a possibility to enter the plateau of the Swabian Jura. In this context, the "Steinlachtal" project was set up to investigate the development of this cultural landscape from the Neolithic until the Iron Age and to analyze the systems of communication used over this time span.

The project has several key requirements for a software-based solution to data management:

- While the research of transport and communication in medieval times could use many different sources, landscape-archaeological approaches for pre-Roman times are mainly based on spatial data, originating from archaeological fieldwork. Therefore, tools were required that could handle geospatial data in a manner that suits archaeological surveys, which requires establishing a stratified system of research contexts from survey site to single find.

- Once the data are collected, one must be able to extract information about the settlement structure, transport routes, preferred places for burials or empty spaces, which were never populated. In comparing the results obtained for one specific period with another, it could be possible to see social dynamics (e.g., moving from the valley ground to the top of hills in unsecure times). Therefore, software with strong support for data analysis and visualization was required.

- To support the research workflow, a high interoperability and the use of accepted standards was desirable. This requires a research environment that not only offers possibilities for collecting (frontend) and storing the data (backend), but also features for extracting the

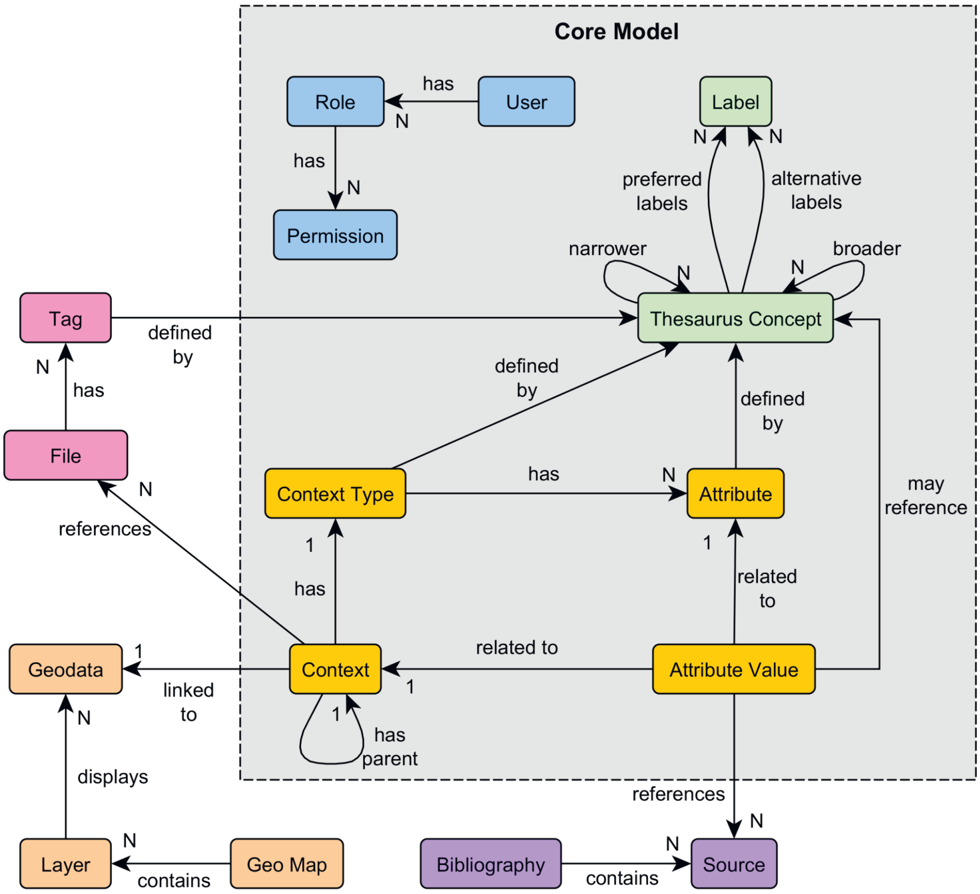

Figure 3. Spacialist data model illustrated as a simplified concept model. The core part is contained in the gray box. The extension parts are outside of the gray box and explained in Section 4.3.

data in standard formats to work with other specialized software for further analyses, e.g. statistical tools like Excel, SPSS or R and spatial analyses in geographical information systems like QGIS.

Spacialist Implementation

System Architecture

Spacialist was built as a web application based on a strictly layered 3-tier architecture separating the client that runs in the web browser, the application programming interface (API) that is exposed by a web server, and the database which is managed on a database server (see Figure 2). The components within these layers build exclusively on open-source tools and frameworks.

The frontend is written in HTML and JavaScript, which allows us to implement an interactive user interface. We use Google's AngularJS framework[3] for most of our JavaScript code, which implements the Model-View-Controller (MVC) (Timms 2016) design pattern to allow separation of concerns when dealing with data, application logic and user interface. This strict separation would even enable third-party vendors to build custom client applications that employ the Spacialist service layer. To simplify the service layer code and for the sake of easier maintenance we use the PHP framework Lumen[4], which can be used to implement RESTful APIs (Masse 2011).

On the backend, all data is stored in a PostgreSQL database[5], which we chose because of its long-standing tradition to adhere closely to the SQL standard. The PostGIS plugin[6] enables the database to support all known spatial reference systems as well as spatial, geometric and geographic objects. This plugin offers full support for geospatial processing and analysis and can also handle virtually any standard format

3 https://angularjs.org

4 https://lumen.laravel.com/

5 https://www.postgresql.org/

6 http://postgis.net/

Attribute Type	Description and User Interaction
string	Text input field
stringf	Textfield for larger strings (e.g. descriptions)
integer	Text input restricted to whole numbers
double	Text input allowing numbers with decimal digits
string-sc	Single-choice dropdown using ThesauRex references
string-mc	Multiple-choice dropdown using ThesauRex references (e.g. tags)
epoch	Combination of year range (e.g. 500 BC – 200 AD); also supports single-choice dropdown for specific era (e.g. Iron Age)
date	Date picker with popup
dimension	Combination of dimensions (length x width x height) and a unit (mm, cm, m, etc.)
geography	Geographical coordinates; can be entered either using WKT or a map popup
percentage	Slider with a range from 0 to 100%
context	A reference to another (existing) context; offered in a dropdown box

Table 1. Currently supported attribute types.

for geodata like GeoTiff, GeoJSON, WKT, KML and many others.

Data Model

As indicated earlier we designed a generic, shared data model that has to be used as a metamodel by all Spacialist project instances. The core of the data model – represented by the gray, dashed-line bordered box in Figure 3 – results in the abstraction that we consider each entity, concept or object within a project domain as a *context* (e.g., the Birkat Al Mouz oasis in the Oman project or the *Silexklinge* (flint blade) find in the Steinlachtal project). Each context can have any number of child contexts, which allows the building of a tree-like hierarchy of domain data (see Figure 4 for an example where *Silexklinge* (flint blade) is child context of the site *Öschingen*).

Each context is of one specific *context type*. The set of context types and their relationships in a project is defined by domain experts (usually the researchers on the project). Each context type consists of a set of *attributes* that are shared by all contexts of this type and capture the contexts' characteristics. Each attribute has a particular value domain, which is defined by assigning one of the available attribute types (see Table 1).

The labels for context types, attributes and enumerated attribute values each reference concepts in the thesaurus that define their localized labels. In the Oman project, for instance, an oasis is described by attributes name (text), type (enumeration), description (text), dating (date range), note (text), and its location (geography). When the researcher creates contexts during the use of Spacialist, each attribute of the context can be assigned a particular *attribute value*.

Multilingualism is achieved using the project thesaurus (see Section 4.4). Labels for context types, attributes and enumerated attribute values each reference concepts in the thesaurus that define their localized labels.

Application Modules

The core data model described above provides the basis for the core functionality of Spacialist, which is implemented in the main application module – the **Context Manager.** This module offers a graphical user interface to create, edit and remove contexts during the project work. The contexts are organized in a tree structure, as depicted by example of the Steinlachtal project main screen in Figure 4, left hand column. When a context is selected in the tree the main screen displays a form to describe the context using the attributes associated with the context's type. For instance, in Figure 4 the Silexklinge (flint blade) context, which has context type Fund (find) was selected and the list of attributes associated with this context type is displayed in the main form. By selecting the star icon next to an attribute's label, the

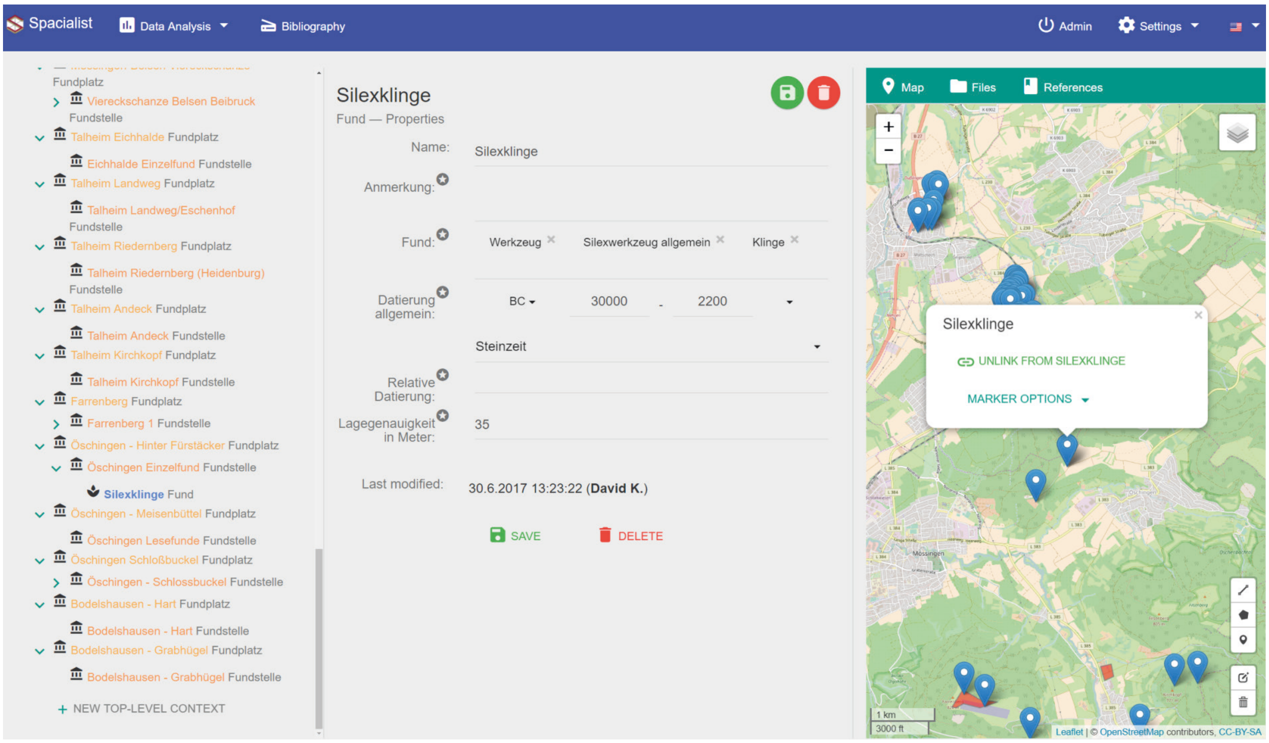

Figure 4. Screenshot of Spacialist instance main screen for the Steinlachtal project.

user can add comments related to the attribute value provided in the main form.

Pivotal for collaborative projects is the **User Manager** module, which is deployed as part of the Spacialist core. Upon each database request, this module checks if a user is logged-in and whether the logged-in user is allowed to invoke a particular request on the service layer, which translates to privileges for adding, updating, and deleting records in the database. User management is implemented using principles of role-based access control (ANSI 2004): Each user can be assigned to one or more roles, and each role comes with a set of permissions (cf. the blue shapes in Figure 3). Each user is then allowed to perform an action, if they are assigned to any role that comes with the necessary permission for the requested action.

All additional functional requirements need to be implemented through extensions of the Spacialist core. This is achieved through add-on modules. Each module may be equipped with a data model that extends the core model, a set of backend services for data processing, and graphical user interface components that implement and offer some well-defined functionality. In the following we describe the existing set of add-on modules.

As a virtual research environment for spatial humanities, Spacialist stores geographical data and allows working with those in the **Geo Map** module. Using PostGIS, all geodata are stored as (multi) points, (multi)linestrings and (multi)polygons within the project's specific spatial reference system. The model is extended to store geo-objects, but also modifies the context database structure. It adds a column for referencing to a specific geo-object. Thereby, it is possible to link each context to a geo-object (see the orange shapes in Figure 3). To interact with these geodata, a map view is added to the user interface that is implemented using Leaflet[7]. When the user selects a context in the Context Manager, its associated geo-object is activated on the map and vice versa. In Figure 4, the geo map is displayed in the right-hand column. Leaflet supports a wide range of map tile sources as base maps, most notably Google Maps, Bing Maps and all kinds of WMS layers[8]. The project administrators may add as many tile map providers as needed by entering the required parameters to fetch the map tiles. Overlays from Web Map Service (WMS) (Geographic information/Geomatics 2005) compliant service endpoints may be added to the map as well. For instance, in the Oman project

7 http://leafletjs.com
8 http://www.opengeospatial.org/standards/wms

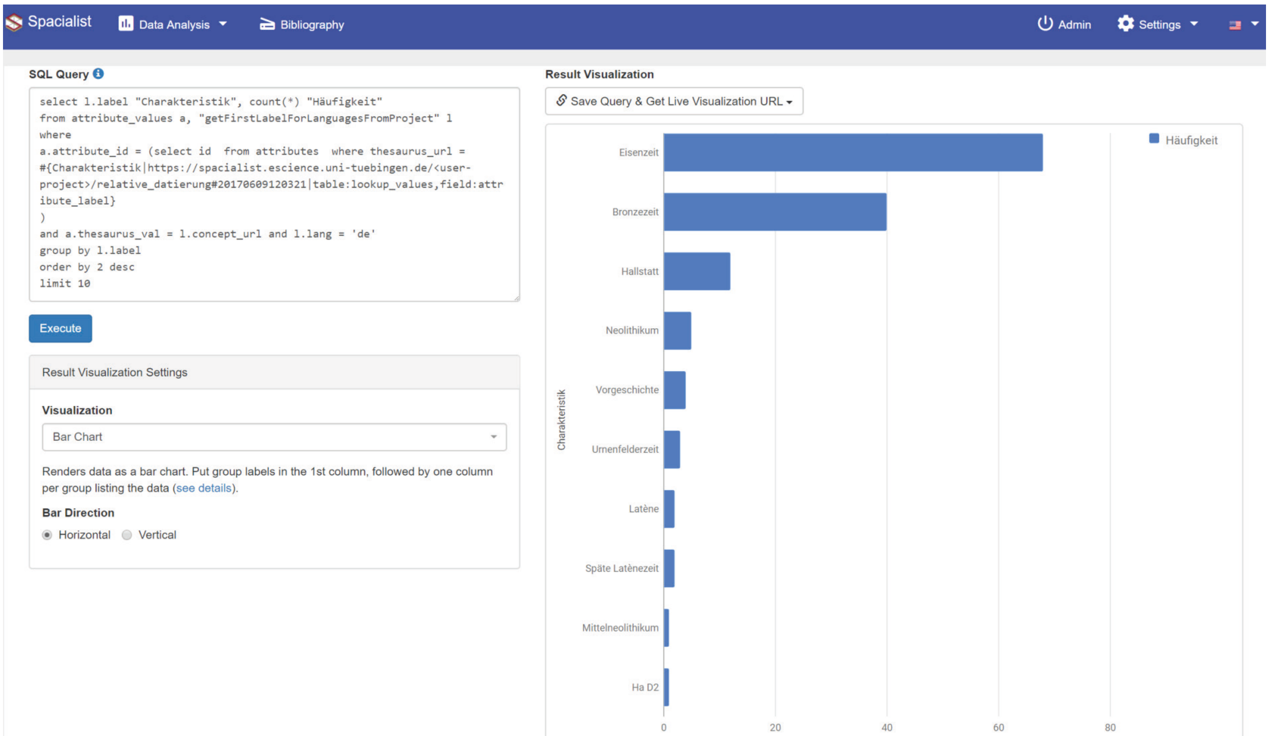

Figure 5. Analysis editor user interface of the data analysis module.

the aerial imagery of the target area was obtained using a high-resolution camera equipped UAV; the resulting images were processed, and the map tiles are now served by our GeoServer[9] via its WMS service. On the map, the geo-objects that reference a context location can easily be modified or deleted. It is also possible to add new markers and geometries on the map (see the edit controls in the bottom right corner of the map in Figure 4). A snapping feature helps to align new edges and vertices of geo-objects with existing ones on the map.

A **File Manager** allows end users to upload and view files, edit their properties, and assign them to contexts (see Figure 3, pink shapes). The add-on extends the core data model by adding a database table to store the relevant file information, amongst others the file URL, modification date and mime type. The file manager also supports the assignment of tags to files. Those tags are used for example to filter files by matching one or more tags. These tags are defined in the controlled vocabulary as part of the project's domain representation (Section 4.4). To simplify file handling, Spacialist can display different file types directly in the browser. This includes 3D viewer files, PDF, markdown, images, audio and video.

9 http://geoserver.org

Spacialist supports bibliography management through the **Bibliography** add-on. The bibliography is stored in a BiBTeX-like (Patashnik 1988) structure directly within the project database. Beside recording entries by hand, end users can import BiBTeX files from other bibliography management tools. Similar to the file manager, this add-on extends the data model to store bibliography data and links to core Spacialist data: each attribute of any context can be associated with references to sources contained in the bibliography (see the purple shapes in Figure 3). The add-on also adds a view to the interface which displays a list of references for selected contexts.

Finally, the **Data Analysis** module allows project team members to formulate queries and query templates to produce data visualizations. As depicted in Figure 5, the analyst enters a parameterized SQL query in the query box and then selects a result visualization which defines how the query result table shall be transformed and displayed to the researchers. Currently the module supports table, bar chart, ankey, timeline, candlestick, map and social network visualizations. Figure 5 shows a query that produces a bar chart with most frequent values for a given attribute. Each query can be stored as a template in the database. The researchers can retrieve any of these

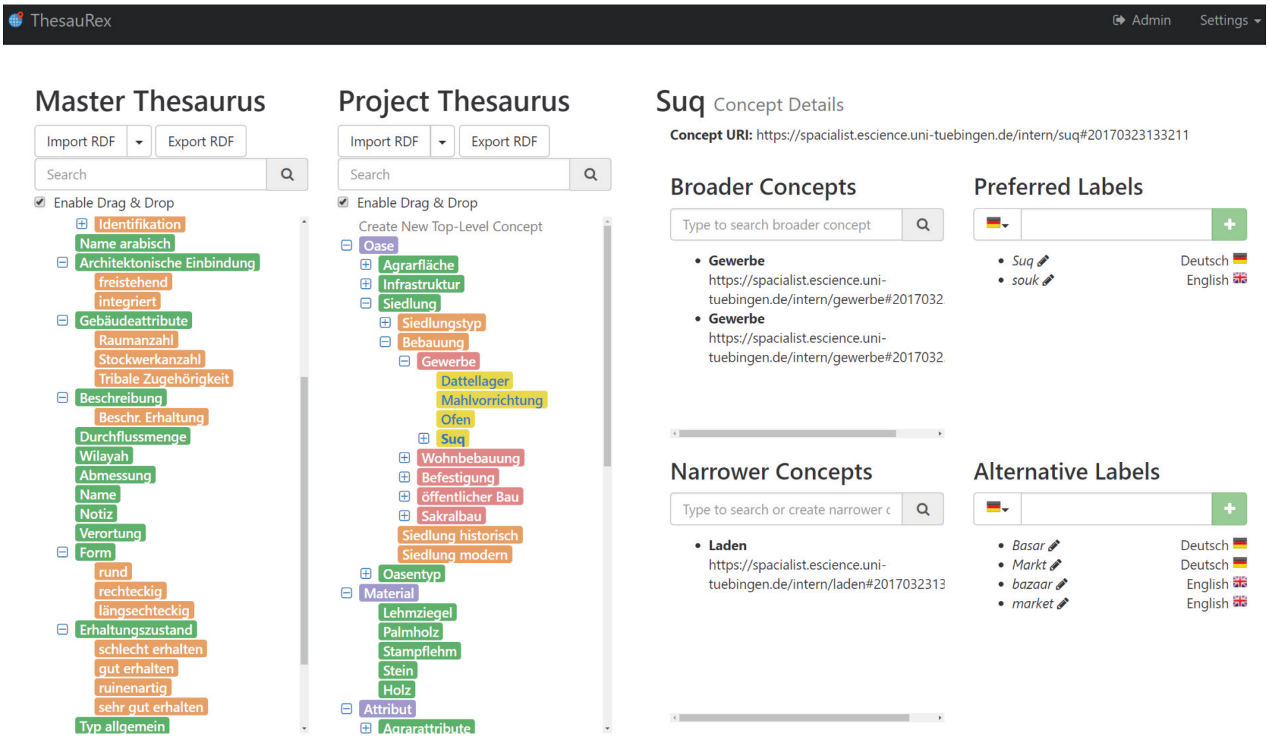

Figure 6. ThesauRex main screen.

stored queries, which will present them with the visualization of cached or live data.

Thesaurus-Based Knowledge Organization—ThesauRex

Employing a thesaurus-based knowledge organization is one of the technical pillars of Spacialist. To increase interoperability and simplicity of use by researchers and domain experts, we decided to adopt the Simple Knowledge Organization System (SKOS)10, a W3C Recommendation first published in 2009. After experimenting with existing thesaurus tools that support SKOS, we found that open-source tools are rare and come with little or no active support. Some proprietary solutions worked well (e.g., the PoolParty Semantic Suite11), but it would have been expensive to buy after the evaluation phase and there was no way to adapt such proprietary tools to specific needs of thesaurus modeling in Spacialist. Therefore, we eventually decided to implement the required subset of SKOS in a tool we called **ThesauRex**. The following description focuses only on those SKOS parts that were implemented in ThesauRex because of their immediate relevance to Spacialist. *Concepts* are the fundamental element in SKOS. A concept can represent basically any entity, object, idea or other item in the domain that is being modeled. They are identified by Uniform Resource Identifiers (URI) and can be labeled in different languages. SKOS supports different types of concept label, most importantly *preferred label and alternative label*. Each concept can have at most one preferred label per language, and the preferred label is used to present the concept to users (e.g., the traditional Arabic market as "souk" in English and "Suq" in German). To allow representation of synonyms, acronyms, abbreviations and other possible labels each concept can be assigned any number of *alternative labels* in any language. For the previous example, alternative labels in English could be "market" or "bazaar" and in German "Markt" or "Basar". The use of alternative and preferred labels offers effective multilingual support also for international collaborations or work groups that have differences in labeling concepts in their domain, which is quite common in many sciences, including archaeology.

To allow organizing the project's knowledge domain, SKOS allows relationships between concepts

10 https://www.w3.org/2004/02/skos/

11 https://www.poolparty.biz/

to be expressed. Any two concepts can be linked using the relationships *broader* and *narrower*. These can be used to represent a graph-based hierarchical knowledge organization. Through transitivity, a graph of *broader/narrower* relationships among concepts can be established. The project vocabularies are structured collections of individual concepts that are representing real world objects, including subjects of research as well as their properties and relationships. These relationships are represented within ThesauRex though an expandable tree structure established by the broader and narrower relationships (see Figure 6). Tree structures are simple to construct from broader/narrower relationships, easy to manage, and arguably more familiar to use and navigate for end users than network structures. In case a concept has a narrower relationship with multiple other concepts, it is displayed as child concept of all these broader concepts.

A unique feature of ThesauRex is its ability to work with not only one but two vocabularies (*cf*. the two concept trees on the left-hand side of Figure 6). While the project-specific vocabulary can be modified by the individual project to satisfy their requirements, the second one may be used as a reference for dragging and dropping parts of the represented knowledge into the project thesaurus. This enables the reuse of domain knowledge within and across research projects. In teams and organizations, the reference thesaurus can also be used as a private or public master thesaurus that is built and refined incrementally and reused by research projects. Both thesauri can be exported and imported via SKOS-compliant RDF files. Therefore, ThesauRex can also be used as a general-purpose, standalone editor for controlled vocabularies, although it was originally designed to be used together with Spacialist to generate and evolve project thesauri.

The experience so far with research projects that use Spacialist reveals that establishing a working data model based on Spacialist's abstract concepts is the most challenging initial task. This task typically requires several working sessions face-to-face between Spacialist team members and the project's researchers to achieve a common understanding of the problem domain and to be able to represent the domain using a SKOS-based thesaurus. For first-time Spacialist users, the initial, agreed project-specific data model is typically implemented by one of the Spacialist experts using ThesauRex. Even for complex domains that include several hundred concepts this can be achieved in less than a day. Previous collaborations with the projects mentioned above have demonstrated that later refinements of the data model, e.g. adding a new context type, or introducing an additional attribute to a context type, or extending the thesaurus for controlled-vocabulary attributes, can usually be performed by the researchers themselves in ThesauRex and Spacialist's data model editor without requiring extensive support.

Conclusion and Outlook

Based on the experiences made in the failed predecessor project ArchGate, it was possible to implement an operational prototype within the first months of the project. This first version is now used in four different research projects. The direct feedback from the researchers using the tool were extremely helpful to implement new functionalities and modules within Spacialist and to adjust the workflow to the needs of the users. The modular structure of Spacialist, the domain-independent data model and the thesaurus-based domain representation facilitated the setup of new projects, which can be done now in a few hours. In addition, adjustments and supplements can be made easily by the researchers within a web interface.

So far, Spacialist has limited data-analysis functionalities. These need to be developed in a next step after a critical mass of research data from the participating projects is available. Another important task will be further integration of the system into the research data repository of the University, so the whole research process from the recording up to the archiving of the data can be covered within a single toolkit. Furthermore, a mapping of reference models like in the CIDOC CRM ontology (ISO 2006) for cultural heritage documentation[12] should be created to guarantee the usability of the data in superior infrastructures.

One of the most challenging, ongoing tasks is the deployment of a sustainable business model, which will allow the long-term operation and the maintenance of Spacialist. To prepare for the transition to

12 See also http://www.cidoc-crm.org

a self-sustaining business model after the project funding ends, Spacialist is already being tightly integrated into the research data management consulting offers that the eScience-Center at University of Tübingen offers to its researchers. The long-term sustainability relies on a minimal set of services to be offered by permanent staff of the eScience-Center. These initially free services include consultation and analysis of the project plan or status, deployment of customized test instances to research projects, and hosting and maintenance of these test instances on University servers. Additional consulting services, long-term hosting commitments, and in particular the development of more complex data models and new project-specific features, are offered as premium services. In most cases currently this means that costs incurred by such services are included as part of the work plan and budget in new project proposals that intend to use Spacialist to manage research data.

Acknowledgements

The project was funded by the Ministerium für Wissenschaft, Forschung und Kunst Baden-Württemberg.

References

ANSI 2004 A*merican National Standard for Information Technology – Role Based Access Control, ANSI INCITS* 359-2004. New York: American National Standards Institute, Inc.

Derntl, M, Renzel, R, Nicolaescu, P, Koren, I and Klamma, R 2015 Distributed Software Engineering in Collaborative Research Projects. In: *IEEE 10th International Conference on Global Software Engineering, ICGSE 2015*. Los Alamitos: IEEE Computer Society, pp. 105-109.

Eve, S and Hunt, G 2008 ARK: A Development Framework for Archaeological Recording. In: Posluschnya, A, Lambers, K and Herzong, I (eds.) *Layers of Perception: Proceedings of the 35th International Conference on Computer Applications and Quantitative Methods in Archaeology (CAA), Berlin, Germany, April 2–6, 2007*. Bonn: Dr. Rudolf Habelt GmbH.

Geographic information/Geomatics 2005 *ISO 19128:2005*. Available at https://www.iso.org/standard/32546.html [last accessed 28 July 2017]

Henze, F, Magdalinski, N, Schwarzbach, F, Schulze, A, Gerth, Ph and Schäfer, F 2013 Concepts and Technologies for a Comprehensive Information System for Historical Research and Heritage Documentation. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Volume XL-5/W2, pp. 325-330.

ISO 2006 *Information and documentation – A reference ontology for the interchange of cultural heritage information* (ISO 21127:2006). Geneva: International Organization for Standardization.

Johnson, I 2011 Heurist: a Web 2.0 Approach to Integrating Research, Teaching and Web Publishing. In: Jerem, E, Redő, F and Szeverényi, V (eds.) *On the Road to Reconstructing the Past. Computer Applications and Quantitative Methods in Archaeology (CAA). Proceedings of the 36th International Conference. Budapest, April 2-6, 2008*. Budapest: Archeaeolingua, pp. 93-99.

Jones, J 2015 *A history of modern Oman*, New York, NY: Cambridge University Press

Lang, M 2012 Datenbank und Geoinformationssystem des Göttinger Agrigent-Surveys. *Göttinger Forum für Altertumswissenschaften*, 15, 2012, 153-166

Masse, M 2011 *REST API Design Rulebook*. O'Reilly Media, Inc.

Patashnik, O 1988 *BibTeXing*. Available at http://ftp.gwdg.de/pub/ctan/biblio/bibtex/base/btxdoc.pdf [last accessed 28 July 2017]

Timms, S 2016 *Mastering JavaScript Design Patterns*. Birmingham, UK: Packt Publishing.

Vogel, O, Arnold, I, Chughtai, A and Kehrer, T 2011 *Software Architecture – A Comprehensive Framework and Guide for Practitioners*. Berlin Heidelberg: Springer-Verlag.