Masterthesis im Rahmen des

Universitätslehrganges „Geographical Information Science & Systems“

(UNIGIS MSc) am Zentrum für GeoInformatik (Z_GIS)

der Paris Lodron Universität Salzburg

zum Thema

Developing a Software-Architecture for Archaeological Raw-Data originating from Excavations in Austria

vorgelegt von:

Mag. Thomas Winklehner

GIS 103467, UNIGIS MSc Jahrgang 2015

Zur Erlangung des Grades
„Master of Science (Geographical Information Science & Systems) – MSc(GIS)“

Gutachter:
Prof. Dr. Josef Strobl

ERKLÄRUNG DER EIGENSTÄNDIGEN ABFASSUNG DER ARBEIT

Ich versichere, diese Masterthesis ohne fremde Hilfe und ohne Verwendung anderer als der angeführten Quellen angefertigt zu haben. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen. Alle Ausführungen der Arbeit, die wörtlich oder sinngemäß übernommen wurden, sind gekennzeichnet.

Hohenzell, am 28.Juli 2017

ACKNOWLEDGEMENTS

First of all I would like to thank my parents for their support during all my academic studies as much as I want to thank my good friend Mag. Johannes Reiter MSc for his counsel in technical terms. Not less do my thanks go to the UNIGIS-Team and - in particular - to my supervisor Prof. Dr. Josef Strobl for making this work possible and giving support in any imaginable way.

CONTENT

ABSTRACT

Since the official Guidelines for Archaeological Measures in Austria [BDA] fail to provide an actual data-model applicable to databases and - by far parts - a binding terminology, archaeological records are given the chance to differ from each other in many structural and semantic ways thus hampering inter-project-data-compatibility. The variety in structure and semantics prevents data-integration and participation in Research Infrastructures, while, at the same time, the habit prevails not to grant other archaeologists access to the collected data in a digital way at all. Thence, this paper illustrates the development of an exemplary software-architecture to overcome the unnecessary structural data-diversification and allow for collective (spatial) data-analysis across campaigns - an endeavour also based on the paradigm of open access to archaeological data. The basic idea here is not to integrate data retroactively, but to accompany the life-cycle of the data with suitable software-tools, in order to facilitate a certain data-structure and terminology for immediate data-compatibility while simultaneously providing mechanisms for data-sharing and publishing. The approach is, by large parts, based on the conception that repetitive tasks recurring with each campaign are better left to automation instead of inventing the same solutions ever anew, so that it appears only reasonable to apply a collective data-model deployed in a bundle of applications. As a result, the effort of data-management will decrease, whereas - aside from various useful tools available by default since originating from database-functions - subsequent data-processing in $G_{eographic}I_{nformation}S_{ystems}$ can resort to suitably prepared data - without the archaeologist having to modify the excavation-process itself.

---

1. INTRODUCTION

1.1. RESEARCH ENVIRONMENT

In order to shed light on the parameters, circumstances and challenges of this paper's topic some recent proceedings on the field of digital-data-management, -sharing and -storing in humanities and archaeology need to be mentioned:

Research Infrastructures

Although the practice of publishing research work via print-media is still very much alive in archaeology - given the fact, that archaeological results tend to be rather long-lasting and some archaeological papers rarely ever obsolesce completely[1] - the shortcomings of analogous media provided nearly exclusively at specialised libraries or research facilities are hardly subject to debate these days any more. However, irrespective of the flourishing of e-journals and e-papers the most relevant trend related to open-access-data as well as long-term-data-storing within recent years is the rise of so called *Research Infrastructures*. [See e.g. Richards 2012; Geser – Niccolucci 2013] Aspöck – Geser describe them as follows:

"Research infrastructures are facilities, resources and services used by the scientific community to conduct research. They include major research instruments, laboratories, databases, libraries and archival collections. Research infrastructures require digital infrastructure and services for information storage, management, processing, publication and access." [Aspöck – Geser 2014]

Since 2002 the *European Strategy Forum on Research Infrastructures* (ESFRI) has been promoting cooperation in policy-making on *Research Infrastructures*, in order to facilitate the so called *European Research Area*.[2] [Aspöck – Geser 2014; ESFRI 03] ESFRI's aim is to overcome fragmentation of data and national policies, so as to provide Europe's researchers with up-to-date *Research Infrastructures*. [ESFRI 01; ESFRI 02;

[1] One may consider papers presenting findings and archaeological records resulting from excavation. This sort of information never really becomes obsolete.

[2] For further information see for example: ERA, HERA, AVROSS, ERIC 01, ERIC 02 or ERIC 03.

ESFRI 03] As a result, several of the latter were either identified or sprung up newly created, as there are in humanities DARIAH (*Digital Research Infrastructure for the Arts and Humanities*) and - more crucial in this context - ARIADNE (*Advanced Research Infrastructure for Archaeological Dataset Networking in Europe*). [Aspöck – Geser 2014] Aside from not referring to archaeology in specific (at the moment) DARIAH-EU is still obviously in the very act of being established,[3] [DAHRIA-EU 01; DAHRIA-EU 02; ESFRI 03] while ARIADNE[4] as virtual e-infrastructure residing as (fourth) level above national/institutional infrastructures and intended to provide a common interface to archaeological data dispersed over several data-providers [Richards 2012; Aspöck – Geser 2014; Geser – Niccolucci 2013] already offers (preliminary) papers on metadata-schemata, that are highly likely to be supported in ARIADNE. Richards [2012] names several rather nationally initiated infrastructures[5] intended to be associated under ARIADNE whereas affirming the impression of lacking comparable archaeological e-infrastructures in Austria.

Thus, any project dealing with data-management, -sharing or -storing in archaeology sure is well advised to take *Research Infrastructures* into account.

Best Practice

Aside from (meta-) data-integration with *Research Infrastructures* archaeological data in Europe needs to fulfill certain requirements - nowadays also in digital terms -, in order to master the latest scientific standards.[6] First of all, in Austria there are the binding, however in terms of data-structure and -semantics not very specific *Guidelines for Archaeological Measures*. [BDA] Fortunately and - with respect european-wide data-integration - beneficially, several projects and institutions already worked out *Guides for Good Practice in Archaeology* and keep them up-to-date while providing them on the internet.[7] Not quite accidentally it is usually in the course of setting up above mentioned *Research Infrastructures* that these guidelines emerge being source of relatively detailed references on how to best handle (digital) archaeological data. As a means to exert collective and qualified data-structure and -semantics across the projects within the infrastructures to some degree, the guides can be an important repository of state-of-the-art qualities to anyone concerned with archaeology.

---

[3] DARIAH-DE, on the other side, does refer to data standards in archaeology (see: https://wiki.de.dariah.eu/pages/viewpage.action?pageId=38080370#Empfehlungenf%C3%BCrForschungsdaten,ToolsundMetadateninderDARIAH-DEInfrastruktur-FachwissenschaftlicheMetadatenstandards%28Content%29 <last visited 15.01.2016>).
Austria's contribution is not specifically archaeological (see: http://www.clarin-dariah.at/de/node/11 <last visited 15.01.2016> and http://www.oeaw.ac.at/acdh/ <last visited 15.01.2016>). Parthenos, however, explicitly comprising archaeology is built around DARIAH (see: http://www.parthenos-project.eu <last visited 15.01.2016> and http://www.oeaw.ac.at/acdh/de/node/312 <last visited 15.01.2016>) and is itself settled in the *Work Program Horizon 2020* [see: http://ec.europa.eu/programmes/horizon2020/en/what-horizon-2020 <last visited 15.01.2016>].

[4] "ARIADNE is an "Integrating Activity" project funded under the Research Infrastructures strand of the *European Union's Seventh Framework Programme* (FP7-Infrastructures-2012-1, contract no. 313193). The project started in February 2013 and will run for four years. It involves 24 partners from 16 European countries who develop a common data infrastructure and related tools, work on knowledge organisation and integration (e.g. use of CIDOC CRM, thesauri, linked data), involve other stakeholders in Europe and beyond, and offer training, good practice guides, and access to their research facilities both on-site and online […]. The overall goal of ARIADNE is to overcome the fragmentation of archaeological data resources and to foster a research culture that values sharing and re-use of data, going beyond organisational and national boundaries. In archaeology, some national data centres, institutional repositories and various other online resources already exist or are under development." [Aspöck – Geser 2014]. Also see Niccolucci – Richards 2013.

[5] In order to enumerate some well known: Archaeology Data Service [http://archaeologydataservice.ac.uk/ <last visited 16.01.2016>], EDNA [http://www.dans.knaw.nl/en/content/categorieen/projecten/edna-e-depot-dutch-archaeology <last visited 16.01.2016>; Gilissen 2013], Swedish National Data Service (SND) [http://snd.gu.se/en <last visited 16.01.2016>], The Strategic Environmental Archaeology Database (SEAD) [http://www.sead.se/ <last visited 16.01.2016>], IANUS [http://www.ianus-fdz.de/ <last visited 16.01.2016>], FASTI Online [http://www.fastionline.org/index.php <last visited 16.01.2016>]. Also see Niccolucci – Richards 2013.

[6] For a general overview see for example DFG 2013.

[7] Archaeology Data Service [http://guides.archaeologydataservice.ac.uk/ <last visited 16.01.2016>]; DARIAH-DE [Puhl 2015, https://wiki.de.dariah.eu/pages /viewpage.action?pageId =38080370 <last visited 16.01.2016> and https://wiki.de.dariah.eu/pages/viewpage.action?pageId=20058160 <last visited 16.01.2016>]; IANUS [http://www.ianus-fdz.de/it-empfehlungen/]; ARIADNE provides references to examples of good practice [see Niven - Weight 2014)].
CARARE is itself a Best Practice Network in order to make content compatible with EUROPEANA. [Richardson 2012].

Since advances in the domain of (applied) informatics, respectively geoinformatics[8], did not leave archaeology unaffected, several technologies have had a deep impact on archaeological research practice and need to be mentioned in the context of this paper, as there are spatial-data-management, 3D-visualisation, WebG$_{eographic}$I$_{nformation}$S$_{ystems}$ and $_{e}$X$_{tensible}$M$_{arkup}$L$_{anguage}$/G$_{eography}$M$_{arkup}$L$_{anguage}$. While today any task handling stochastic archaeological data usually relies on data-management-tools like databases,[9] the benefits of spatial-data-management-systems, respectively spatial databases, as the backbone of spatially enabled applications have already been recognized and exploited by a number of projects,[10] among which there are more than a few archaeological WebGIS-applications.[11] The latter - involving traditionally 2D representations of structures and/or locations - have been developed into 3D-enabled applications recently,[12] in order to suffice the purpose of visualising even above-ground ruins and buildings. The function of archaeological WebGIS typically rests with simple localisation of sites, display of excavation-findings, virtual-reconstructions or rendering the locations of cultural heritage.[13] The data-management on the excavation-site, however, is commonly left to the responsibility of the archaeologist in charge (albeit he can resort to a few ready-to-use software solutions)[14].

It is estimated that at the moment only 6 to 8% of all researchers in archaeology give free access to their data, so that data-loss impends. [Aspöck – Geser 2014] In this regard it is crucial to note that, although rescue-excavations oblige the causer to take financial responsibility for the required archaeological measures [Denkmalschutzgesetz], the major financial means are supplied by public funds[15], not to mention the public interest in cultural heritage. [see for example Denkmalschutzgesetz]. So even though sites of cultural heritage must be preserved and, thus, in some cases information about them kept secret, archaeological raw-data may be considered open data in a certain sense.[16]

1.2. General Objectives

Upon having given a rough summary on the current situation of data-management within Europe's archaeology in the course of the previous chapter, important effects of technologies and recent trends on the archaeological research field were revealed having apparently led towards a multitude of solutions, data-set-types [Costa a.o. 2014] and even languages employed [Niccolucci – Richards 2013].[17] Isaksen and others [2009] put it the following way: "Whilst the datasets all pertain to the same domain, they frequently employ mixed taxonomies and are heterogeneously structured…", which is probably rooted in archaeological vocabularies having a

[8] see for example Brovelli – Magni 2003 or Scianna – Villa 2011.
[9] Observation of the author; also see Meyer and others [2006] for a little overview on the application of technologies in archaeology until 2006.
[10] see for example Apollonio and others 2010.
[11] see for example Sebillo and others 2003; Semeraro 2007; Mantegari – De Salvo 2008; Cefalo and others 2011; Gallo – Roberto 2012; von Groote-Bidlingmaier and others 2015.
[12] see Richards-Rissetto and others 2010; De Luca and others 2011; Billen and others 2013; Herban – Vilceanu 2013.
[13] Archaeological strata hardly ever appear to be subject to 3D WebGIS-mapping.
[14] The website http://www.bajr.org/bajrresources/software.asp <last visited 11.10.2015> lists all sorts of archaeologically useful software. On-site data-management applications can be found at: http://ark.lparchaeology.com/about/ <last visited 11.10.2015>, http://www.arctron.de/de/produkte/software/archaeodata/ <last visited 11.10.2015>, http://datalino.de/nerik/ <last visited 11.10.2015>, https://www.esdm.co.uk/hbsmr-historic-environment <last visited 11.10.2015>, https://www.inari-software.com/?lang=de <last visited 11.10.2015>, http://www.iadb.org.uk/ <last visited 11.10.2015>, http://www.intrasis.com/engelska/index_eng.htm <last visited 11.10.2015>, http://www.amh.tg.ch/xml_74/internet/de/application/d813/f820.cfm <last visited 11.10.2015>, http://www.nabonidus.org/ <last visited 11.10.2015> and http://www.singularch.de/pocket_e.htm <last visited 11.10.2015>.
In order to give a rough overview of the explored software solutions, several generally observed shortcomings need to be addressed: The applications tend to be solely web-based, not referring to metadata-standards, best-practice-guides or archiving-friendly document-types and manage alphanumeric and spatial data separately.
[15] see for example: https://wissenschaft.bmwfw.gv.at/bmwfw/forschung/national/forschungseinrichtungen/das-oesterreichische-archaeologische-institut-oeai/ <last visited 13.05.2017>.
[16] see https://www.data.gv.at/infos/zielsetzung-data-gv-at/ <last visited 13.05.2017> or http://www.inspire.gv.at/ <last visited 13.05.2017>.
[17] National endeavours and approaches have hardly brought benefit in terms of comparability of data or knowledge on a european level. [Niccolucci – Richards 2013].

tendency to be only maintained on an organizational level. [May and others 2011] So with Karagiannis and others [2012] it can be stated that "…multilingual knowledge representation, access and translation are an impending need."

On the other hand, future research is expected to be technology- and big-data-driven [Riding the Wave 2010; Geser – Niccolucci 2013; Aspöck – Geser 2014] therefore transferring the more emphasis from national or institutional data-stores towards international solutions. [Geser – Niccolucci 2013; Niccolucci – Richards 2013; also see: Riding the Wave 2010 and Unsworth 2006] In order to meet these issues and integrate all kinds of data from miscellaneous sources properly, *Research Infrastructures* like ARIADNE are being established though raising the question to what degree diverse data must and can be integrated at all. The current practice in cultural heritage and especially in ARIADNE, thereby, is either to incorporate established metadata-standards or (- what appears to be the prevailing habit -) map metadata according to an accepted pattern, more precisely in congruency with an ontology providing semantics, for reusability of datasets needs to meet certain criteria in terms of purpose, exactness and reliability expressed by metadata.[18] [Geser – Niccolucci 2013; for ARIADNE's practice of mapping in accordance with CIDOC-CRM see ARIADNE or http://139.91.183.82:8888/drupal/use-cases <last visited 16.10.2015>]

1.2.1. Hypothesis and Proposition

As far as in this very regard archaeological excavation-data is concerned, the present paper intends to bring together the major points[19] discussed in the preceding chapters by presenting the development process of a software-architecture as a means to accompany - in fact, facilitate - the full life-cycle of archaeological raw-data originating from excavation, so that eventually inter-campaign-compatible data meeting accepted standards is produced. Special interest rests, of course, with the spatial aspect in all regards.

The idea behind this approach is that, while aforementioned mapping-process usually integrates diversely structured data of variant meaning on a rather abstract semantic meta-level, Austria's excavation-data already is subject to regulation due to national guidelines[20] - one may nearly regard as first step towards a data-model - and could, therefore, be combined effortlessly on a national basis applying a collaborative data-model and application. In order to stress the normalised-, stochastic- and inventory-like-character of excavation records, it suffice to remark that the excavation process nowadays has a habit of being outsourced to specialised, however commercial companies. As an archaeologist or excavation-professional is primarily an expert in his domain, though, rather than an IT-specialist and does not need to be overly burdened with setting up data-management-tools for various campaigns ever anew - especially in the light of already existing guidelines -, this work aims for developing a comprehensive application-architecture capable of employing such data-structure.[21]

---

[18] Although it is far beyond the scope of this paper to fully discuss the idea of the semantic web [see e.g. Berners-Lee and others 2001], it is worth mentioning that its consequences reached cultural history particularly in the form of ontologies (e.g. CHARM [www.charminfo.org <last visited 16.10.2015>; Costa and others 2014]), the most popular of which is CIDOC-CRM by now established as ISO Standard (21127:2014) [http://www.cidoc-crm.org/ <last visited 16.10.2015>; http://www.iso.org/iso/catalogue_detail?csnumber=57832 <last visited 16.10.2015>]. While the most common ontology in cultural heritage is CIDOC-CRM [see Doerr and others 2004; Szász and others 2006; Eide and others 2011; Karagiannis and others 2012; Costa and others 2014], there are a number of (partly proprietary) elaborate metadata-standards like LIDO and *Europeana Data Mode*l (EDM), both of which were heavily influenced by CIDOC-CRM. [Geser – Niccolucci 2013 with overview] LIDO and CIDOC-CRM are sure not satisfactory for archaeological tasks, but they are flexible and extensible. [Geser – Niccolucci 2013] Two specialised archaeological extensions to CIDOC-CRM worth mentioning have also been applied in practice: 1. CIDOC-CRM-EH [see STELLAR; May and others 2011] 2. $CRM_{archaeo}$. Extensions, however, are only useful when commonly agreed on and recognized.
Also see Costa and others 2014 for a different attempt to integrate archaeological metadata.

[19] …namely: ability for integration in *Research Infrastructures*, conformity with Austria's guidelines on archaeological measures, guides on good practice, archivable datatypes, on-site data-management-tools and WebGIS.

[20] The core set of data-fields within excavation-datasets is generally heavily determined by national guidelines and usually resemble a kind of inventory. For Austria see BDA.

[21] "Automatic and easy to use tools are also necessary for the mainstream uptake of the semantic web in the cultural heritage community." [Karagiannis and others 2012].

So in contrast to *Research Infrastructures*' strategy to integrate existing cross-domain- and cross-origin-datasets retroactively the intention here - on the one hand - is to provide a software-tool, that not only enables the archaeologist to produce valid (spatial-) data in terms of structure and semantics comprehensible in the sense of aforesaid guidelines from the very beginning of an excavation-project onward, but also preserves the full potential of data structured uniformly across campaigns in accordance with recognised standards[22]. On the other hand, the tool will manage to pool together different data-sets - as long as they adhere to the same data-model - and make them accessible via website and $W_{eb}F_{eature}S_{ervice}$, so as to lend itself to a resource for *Research Infrastructures*. As a result, the produced records also fit well in between the open data-sets (*Open Government Data*)[23] already provided by (federal) states in Europe,[24] so that the local reference framework for this paper's case study shall be Austria.

1.2.2. General Requirements

As a means to elaborate the functional needs to be met by the software a task commonly applied is identifying use-cases, from which requirements are deduced. [see UML Glasklar 2012] Since those very use-cases are later employed to evaluate this project's yields, the corresponding illustrations can be found in chapter *3.1. Verifying Results.* What follows are several important issues worth of closer discussion.

Data-Model

In the spirit of the foregoing paragraphs adumbrating the need for a data-model this endeavour of software-development requires the elaboration of a data-model-scheme describing a valid model-structure accepted by the software. The scheme as well as the resulting data-models must be readable, editable (in case of the models), ready for archiving and easily exchangeable between systems and people. In case the data-store is intended to be affiliated to a *Research Infrastructure*, the models must also be capable of carrying an ontology's entities and predicates together with its model-items.

Glossaries / a-priory-Vocabularies

As soon as classifying is involved, the demand for standardised descriptive terms arises leading towards this software's requirement of supporting some kind of glossary. Because such a-priory-vocabularies may be regarded as part of the data-model, they, too, are subject to the same standards in terms of readability, editing, archiving and exchange.

Data

Although archaeological projects tend to utilise technologies, such as surveying-instruments, in order to generate digital datasets directly, it is still mostly the human expert creating and assembling data thus representing the performance-bottleneck even for automatically- or device-generated data being joined with manually recorded information anyway. Thence, the prevailing requirement here is the traditional and still predominantly employed practice of humans inserting data into the digital medium. Beside the common data-management capacities special emphasis must rest with handling geometries, durable storage of non-binary data as well as making the records accessible via approved and widely accepted gateways. Binary files, however, of miscellaneous size containing, for instance, photos or 3D-models must be manageable together with alphanumeric data. Metadata is not a big concern in regard of the program itself, because it can always be provided as a designated part of the data-model.

Assessment

In order to absolutely clarify the application's task, it must be stated that the main requirement is not to develop a plugin, extension or converter to an

---

"Mapping requires skills and knowledge which are uncommon among the cultural heritage professionals with the most thorough knowledge of the material to be mapped. It is not only the absence of facilitating tools, but also the existence of practices, legal obligations and the lack of a clear motivation that has as yet delayed or reduced the creation of such mappings to a handful cases." [Eide and others 2011]

[22] In this context 'standards' refers mostly to geometry not being published in proprietary, but in open format.

[23] The open datasets can in return be used in an application. For Open Data see, for example, https://www.land-oberoesterreich.gv.at/opendata.htm <last visited 18.10.2015>.

[24] Restricted access may be necessary due to protection of cultural heritage.

existing $G_{eography}I_{nformation}S_{ystem}$ or database-system, but the true backbone of excavations' data-handling, which itself may in future be subject to extension.

Operating Systems

The gained knowledge and procedures of this endeavour must be somewhat universally applicable in the regard that the software-architecture can be adapted to run on the three main operating systems of personal computers. Additionally, referenced functions-libraries must not be proprietary, for they shall be used freely. A desirable characteristic would also be the software not requiring installation.

Special Conditions

As the planned software is supposed to be tailored for deployment in excavations, this venture not only must pay tribute to certain peculiarities of the archaeological field of application, but can also take advantage of that circumstance. First of all, archaeological field studies are organised in campaigns and projects in such a way as to enable data-handling to be grouped by those endeavours. When querying information, for example, only data from a certain campaign may be involved. In addition, once archaeological records are gathered, they rarely experience change as, for instance, a warehouse-inventory would do. On the other hand, during the recording process several people must be able to access and insert data simultaneously, while, at the same time, records may have to be transferred. Even though the internet can not be relied on as medium for data-transfer in the project phase, the project-staff usually works at the same location anyway - thus facilitating networking via local files-exchange. Additionally, a simple backup-mechanism might be beneficial. In general, simplicity in handling is regarded a virtue given non-IT-specialist-archaeologists.

Scale

Even though several aspects on the issue of the long-term storage of digital data may be covered in assembling the software-architecture, that topic is not supposed to be a central point, because the software's domain is intended to reach from single user to project- or even inter-project-scale, however not to the wide institutional level. While the resulting data is definitely aimed at contributing to big institutional data-stores, the software itself is not supposed to fulfill the job of a large-scale repository. Scaling efficiently within its field of application, though, is a desired effect.

1.2.3. Conceptual Solution

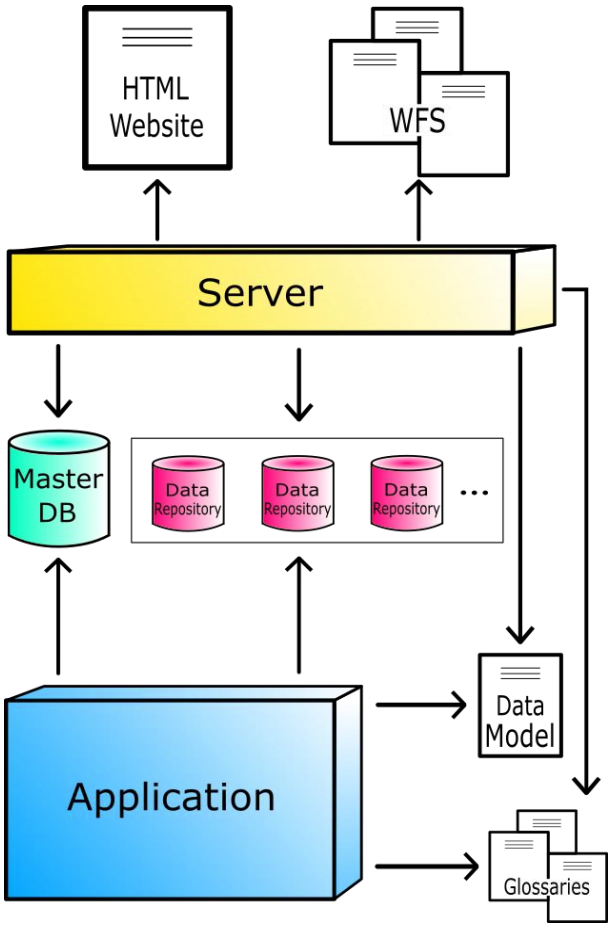


Figure 1

simple schema

Representing the most basic and significant part of the architecture the choice on the database-system impacts all major components (as illustrated by Figure 1) and shall be discussed first. Among the commonly deployed non-proprietary spatial database-systems[25] there is, above all apparently, the widely known PostgreSQL/PostGis[26]-system [Brinkhoff 2013],

[25] for a brief overview see Brinkhoff 2013.
[26] https://www.postgresql.org/ <last visited: 10.02.2017>.

which unfortunately does not comply well with the requirements for file-based storage and portability, as it generally appears to be interlaced with the machine it runs on to non-negligible degree.[27] Its comparatively heavy footprint on the system - due to its amount of functions and storage-ability -, though, make it more of a candidate for institutional-level data-storage (and can still be integrated to the application in future endeavours). A suitable match to the current requirements, however, is the SpatialLite-database-system[28] with its strictly file-based-paradigm happily capable of being integrated with mobile applications. [Brinkhoff 2013] Although limited[29] in storage-capacity compared to server-like systems as aforementioned PostgreSQL/PostGis, its known limitations seem insignificant when storing campaign-sized archaeological alphanumeric data only. Likewise, as outlined earlier data-entry is supposed to be accomplished by humans thus restricting data-flux-density to a degree within human scale. As the requirements-chapter indicated (see *1.2.2. General Requirements*), binary data does not need to be stored in the database - given the fact, that typical database-tasks usually involve alphanumeric values and geometry -, but has to be found if needed. Besides, there might be a size-limit to so-called BLOB-fields in a database-system.

So in summary, the main characteristic of the architecture is its employment not of a single database, but of a swarm of database-files each representing, for instance, an excavation-project, which pattern not only corresponds to the campaign-like archaeological field-research paradigm and the requirements outlined, but also brings certain challenges to the application (together with advantages to the undertaking as a whole). As illustrated by Figure 1 and Figure 2 the main executable component for the local user is a desktop-application residing above the database's $A_{pplication}P_{rogramming}I_{nterface}$ managing all data-handling tasks. In order to work correctly, it accesses the data-model as well as the glossaries, which both have to be on hand as separate text-documents. A master-database is needed for permanent storage of application-relevant information such as file-paths. The named documents as well as the master-database are also consulted by the server-component making the file-system the only connection between two separate applications: the desktop-program and the server. The separation, thereby, prevents problems in applying the architecture on different operating systems and makes the server-part an optional, exchangeable feature. The server - constituted by an existing free-to-use software-solution - is entrusted with publishing information about the data-model, the glossaries and the data itself on a website, so that a remote user gets an impression on what the data-store provides. The data itself shall be retrievable by the user primarily via service, namely $W_{eb}F_{eature}S_{ervice}$[27], so that it can be requested in accordance with a specified standard.

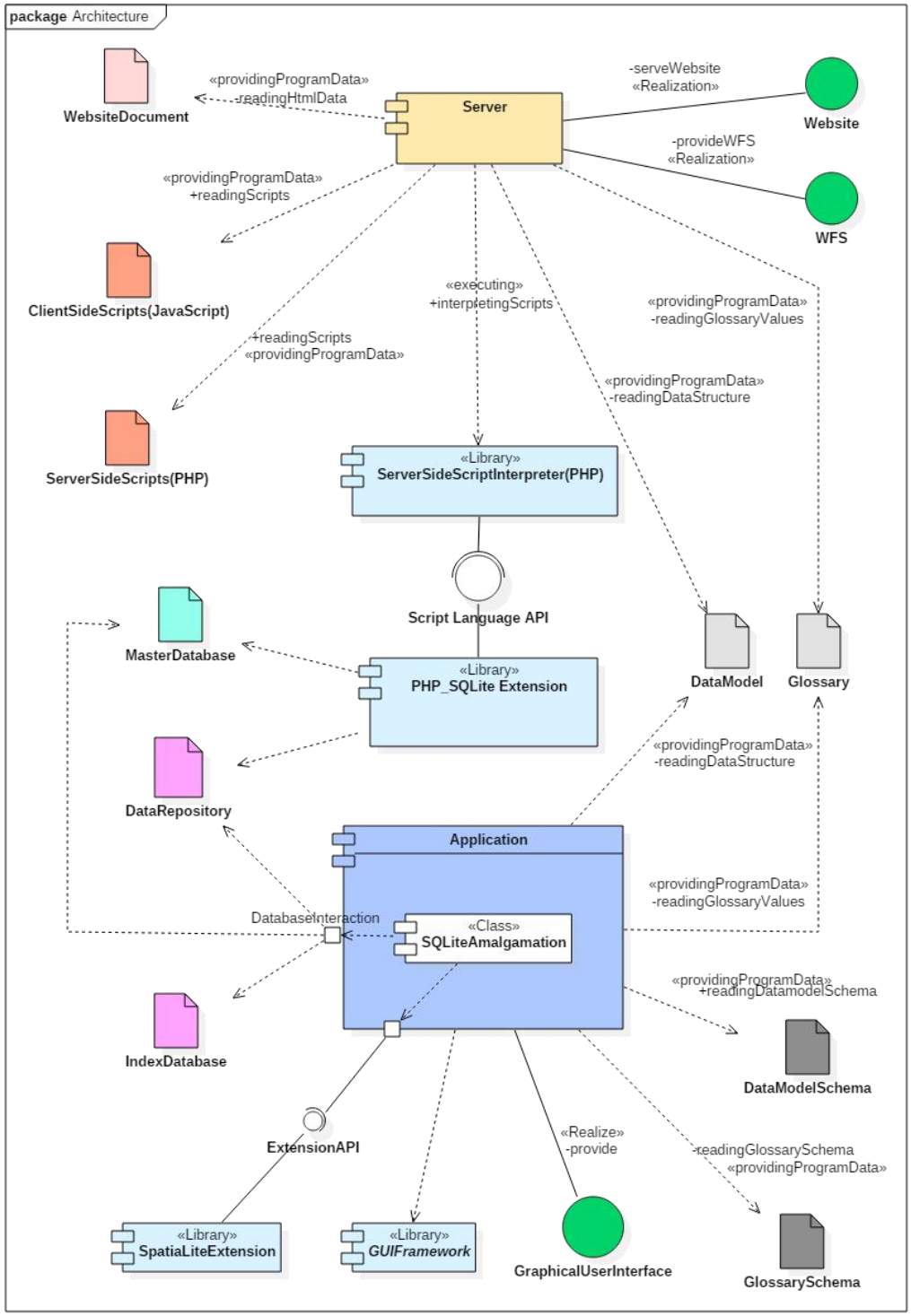
In regard of technologies applied and considering the requirement of portability the programming language Java and its runtime-environment[28] may seem a reasonable foundation, as, on the one hand, the source-code for the desktop-application remains static irrespective of the operating system, on the other hand the popular server Apache Tomcat[29] as well as the $W_{eb}M_{ap}S_{ervice}$/$W_{eb}F_{eature}S_{ervice}$-server named Geoserver[30] rely on the same technology anyway. The dependency on a runtime-environment, however, is undesirable especially when applying a software supposed to work without internet-connectivity, so that technologies running natively on the operating system are preferred.[31]

[27] A portable, however still a little overly complicated distribution is available at: https://sourceforge.net/projects/postgresqlportable/ <last visited: 10.02.2017>.
[28] https://www.sqlite.org/<last visited: 10.02.2017>; Owens 2006.
[29] https://www.sqlite.org/limits.html<last visited: 10.02.2017>.

[27] http://www.opengeospatial.org/standards/wfs<last visited: 11.02.2017>; also see for example: Brinkhoff 2013 or Tsalgatidou – Pilioura 2002.
[28] see: https://java.com/de/<last visited: 11.02.2017> or https://www.oracle.com/de/index.html<last visited: 11.02.2017>.
[29] see: http://tomcat.apache.org/<last visited: 11.02.2017>.
[30] see: http://geoserver.org/<last visited: 11.02.2017> or Youngblood 2013 and Henderson 2014.
[31] One may imagine a scenario, in which archaeologists are supposed to be equipped with this software, but need to install a runtime-environment first. However, due to the remote excavation-site and the lack of internet-connectivity the endeavour fails.

In this respect it is not only worth noting that SQLite[32] is written in C and has a C/C++ $A_{pplication}P_{rogramming}I_{nterface}$[33], but also that Windows, MacOS as well as Linux all go back to the C-programming language making C, respectively C++, the first choice for creating the desktop application.

In terms of web-technologies there is hardly a way around HTML, CSS and JavaScript [see Nixon 2014], whereas server-side scripting goes hand in hand with the selected server's capabilities. A very widely employed language, though, is PHP [Tatroe and others 2013], which is, thus, an excellent candidate for server-side scripting and allows to select from a range of servers, as there is for example the XAMPP-bundle[34].



It is worth noting that, on the whole, the general requirement to the software-architecture is enforcing and supporting the data-model throughout all its functionality, the fulfillment of which may be considered the confirmation of success.

Applying a content-management-system, a web-framework like Django or cloud-hosting is explicitly avoided here, because this paper intends to exemplify the implementation under the use of the most basic web-technologies.

Figure 2

UML2-compiant component-diagram

[32] see: https://www.sqlite.org/<last visited: 12.02.2017>.

[33] see: https://www.sqlite.org/cintro.html<last visited: 12.02.2017>.

[34] see: https://www.apachefriends.org/de/index.html <last visited: 11.02.2017>.

2. IMPLEMENTATION: METHODS AND RESULTS

The following sub-chapters review the technical aspects of the software-architecture together with its development process. As this is the very core task of this endeavour according to the definition given above (see *1.2.1. Hypothesis and Proposition*), it is designated the most extensive treatise in this paper. The attentive reader may thereby notice the huge amount of effort needed to run into setting up the applications' general business logic, before spatial functions can be explicitly approached and implemented. In this respect, however, one must not forget that the overall undertaking itself is - above all - motivated by the goal of handling and finally publishing spatial data.

Because the development is done on a windows 10 operating system, all information given on the remaining sites is valid for exactly that system by default. Adaptations necessary for application on different systems are hinted when required.

The sub-chapters, respectively sub-topics, are roughly sorted in an order representing development-steps from the most basic to the most advanced beginning with the data-model and the vocabularies, passing the desktop-application, in order to finally treat of the server with its website and the WebFeatureService.

2.1. GENERAL REMARKS ON THE SOFTWARE DEVELOPMENT PROCESS

When considering software development the question soon arises of which method, 'school' or tradition to follow. In reference to a typical guide on that topic [see for example Dooley 2011] there are a few important remarks to take, in order to highlight the structure of this development-process:

First of all, the author of this text impersonating the sole developer is neither a professional programmer nor had ever experienced the practical aspects of large-scale software-development before the beginning of this endeavour hence making that circumstance part of the case-study while making guidebooks - usually building up on some degree of experience - harder to apply.

The approach settled for, consequently, was to enforce a modular and strictly bottom-up strategy of consistently stacking higher-level components atop rather basic elements. Unlike an analytic top-down approach the policy applied here does not seek to anticipate the full detailed application-architecture in advance - for that would presuppose a high amount of practical experience - , but is merely driven by adding piece by piece towards an endpoint. Scheduling stopovers and setting intermediate objectives was therefore rather motivated by a strategy than by a detailed preconceived road map. Besides, as one may conclude from Dooley [2011], the general vision of the finally functioning software-structure is not infrequently put into practice via more of an iterative gradual refinement as much as the resulting components, whether standing alone or having already been assembled, require careful testing thence dedicating any software-composing-process to some kind of repetitive test-driven pattern.

So in order to roughly characterize the development's actual proceedings the following shall be stated: Components, modules, objects, functions etc. were created and tested as individual items, if possible, before incorporated into the software-structure, where in-context testing happened. So until in the course of that process the applications' final state was reached, test-scenarios had to be engineered.

2.2. Data-Model

As a means to meet the data-model-related requirements, the most basic task is to discuss the most suitable document type, which, for once, needs to be readable for humans, thence, textual. Taking into account the remaining requirements, the perfectly matching technology reflecting the necessary qualities emerges from the various file-formats in the form of eXtensible Markup Language. [see: https://www.w3.org/XML/ <last visited: 14.02.2017> and Vonhoegen 2013] It not only allows for customly structuring documents, but also to define that structure in a textual schema hence informing the user about the scope of validity, while conversely facilitating validation against the scheme. [see: Van der Vlist 2002]

2.2.1. Scheme

```
<?xml version="1.0" encoding="UTF-8"?>

<!--  Author: Mag. Thomas Winklehner
  Purpose/Description: Schema-Document for Databasestructures
  Date: 21.11.2015
  Warning: DO NOT MODIFY THIS DOCUMENT
-->

<!-- using XSD 1.0 -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.thomaswinklehner.at/ArchaeologicalDatabase"
  xmlns="http://www.thomaswinklehner.at/ArchaeologicalDatabase"
  xmlns:tw="http://www.thomaswinklehner.at/ArchaeologicalDatabase"
  elementFormDefault="qualified">

 <!-- Outlines for the Database Design -->
 <xs:element name="Database">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="Table" minOccurs="0" maxOccurs="100">
     <xs:complexType>
      <xs:sequence>
       <xs:element name="Column" minOccurs="0" maxOccurs="100">
        <xs:complexType>
         <xs:sequence>
          <xs:choice>
           <xs:element name="DataColumn">
            <xs:complexType>
             <xs:sequence>
              <xs:element name= "DataColumnType" type="DataColumnTypeType"/>
              <xs:element name="Constraint" type="ConstraintType" minOccurs="0"
                    maxOccurs="5"/>
              <!-- For Descriptions -->
              <xs:element name="ColumnDescriptionGerman" type="xs:string"/>
              <xs:element name="ColumnDescriptionEnglish" type="xs:string"/>
             </xs:sequence>
            </xs:complexType>
           </xs:element>
           <xs:element name="SpatialColumn" type="SpatialColumnType"/>
          </xs:choice>

          <!-- Ontology-Items -->
          <xs:sequence minOccurs="0" maxOccurs="1">
           <xs:element name="OntologyPredicate" type="xs:string"/>
           <xs:element name="OntologyEntity" type="xs:string"/>
          </xs:sequence>
```

```
      </xs:sequence>

      <!-- Table Name needs to be repeated for referencing Foreign Key -->
      <xs:attribute name="TableName" type="NameType" use="required"/>
      <xs:attribute name="ColumnName" type="NameType" use="required"/>
     </xs:complexType>
    </xs:element>

    <!-- For Descriptions -->
    <xs:element name="TableDescriptionGerman" type="xs:string" minOccurs="1"
          maxOccurs="1"/>
    <xs:element name="TableDescriptionEnglish" type="xs:string" minOccurs="1"
          maxOccurs="1"/>

    <!-- Ontology-Items -->
    <xs:choice minOccurs="0" maxOccurs="1">
     <xs:element name="OntologyPredicate" type="xs:string"/>
     <xs:element name="OntologyEntity" type="xs:string"/>
    </xs:choice>

   </xs:sequence>

   <xs:attribute name="TableName" type="NameType" use="required"/>

   <!-- Purpose of the Table: for storing or providing Terms -->
   <xs:attribute name="Use" use="required">
    <xs:simpleType>
     <xs:restriction base="xs:string">
      <xs:enumeration value="Storage"/>
      <xs:enumeration value="Vocabulary"/>
     </xs:restriction>
    </xs:simpleType>
   </xs:attribute>
  </xs:complexType>

  <!-- Making Table-Name Reference for Use with the Columns of exactly that Table -->
  <xs:key name="UniqueTableName">
   <xs:selector xpath="."/>
   <xs:field xpath="@TableName"/>
  </xs:key>

  <!-- referencing Table-Name  -->
  <xs:keyref name="UniqueTableNameRef" refer="UniqueTableName">
   <xs:selector xpath="tw:Column"/>
   <xs:field xpath="@TableName"/>
  </xs:keyref>

  <!-- Making ColumnName unique in the Scope of a Table -->
  <xs:key name="Columns">
   <xs:selector xpath="tw:Column"/>
   <xs:field xpath="@ColumnName"/>
  </xs:key>

 </xs:element>

 <!-- Specify Indices  -->
 <xs:element name="Index" minOccurs="0" maxOccurs="unbounded">
  <xs:complexType>
```

```
     <xs:attribute name="Name" type="NameType" use="required"/>
     <xs:attribute name="Unique" type="xs:boolean" use="required"/>
     <xs:attribute name="TableName" type="NameType" use="required"/>
     <xs:attribute name="ColumnName" type="NameType" use="required"/>
    </xs:complexType>
   </xs:element>

  </xs:sequence>
 </xs:complexType>

 <!-- Making Table-Name unique in the Scope of the Database-->
 <xs:key name="UniqueTableNameInDataStructure">
  <xs:selector xpath="tw:Table"/>
  <xs:field xpath="@TableName"/>
 </xs:key>

 <!--Making TableName and ColumnName unique and Key in the Scope of a Database-->
 <xs:key name="PrimaryKey">
  <xs:selector xpath="tw:Table/tw:Column"/>
  <xs:field xpath="@TableName"/><!--Tablename must be repeated for Foreign Key-->
  <xs:field xpath="@ColumnName"/>
 </xs:key>

 <!-- Making Index-Table-Column-Pair unique in the Scope of a Database -->
 <xs:key name="IndexTableColumnKey">
  <xs:selector xpath="tw:Index"/>
  <xs:field xpath="@TableName"/>
  <xs:field xpath="@ColumnName"/>
 </xs:key>

 <!-- Making IndexName unique in the Scope of a Database -->
 <xs:key name="IndexNameKey">
  <xs:selector xpath="tw:Index"/>
  <xs:field xpath="@Name"/>
 </xs:key>

 <!-- Referencing TableName and ColumnName to Indices  -->
 <xs:keyref name="IndicesTablesColumnsRef" refer="PrimaryKey">
  <xs:selector xpath="tw:Index"/>
  <xs:field xpath="@TableName"/>
  <xs:field xpath="@ColumnName"/>
 </xs:keyref>

 <!-- Referencing TableName and ColumnName to Foreign Key  -->
 <xs:keyref name="ForeignKeyRef" refer="PrimaryKey">
  <xs:selector xpath="tw:Table/tw:Column/tw:DataColumn/tw:Constraint/tw:FOREIGNKEY"/>
  <xs:field xpath="@RefTableName"/>
  <xs:field xpath="@RefColumnName"/>
 </xs:keyref>

</xs:element>

<!--***************Column-Datatypes******************-->

<!-- DataColumnTypeType -->
<xs:complexType name="DataColumnTypeType">
 <xs:choice>
  <xs:element name="NVARCHAR">
   <xs:complexType>
```

```
    <xs:attribute name="Length" use="required">
     <xs:simpleType>
      <xs:restriction base="xs:positiveInteger">
       <xs:pattern value="[0-9]{3}"/>
       <xs:minInclusive value="2"/>
       <xs:maxInclusive value="200"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
   </xs:complexType>
  </xs:element>

  <xs:element name="NUMERIC">
   <xs:complexType>
    <xs:attribute name="Precision" use="required">
     <xs:simpleType>
      <xs:restriction base="xs:positiveInteger">
       <xs:pattern value="[0-9]{2}"/>
       <xs:minInclusive value="1"/>
       <xs:maxInclusive value="20"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="Scale" use="required">
     <xs:simpleType>
      <xs:restriction base="xs:nonNegativeInteger">
       <xs:pattern value="[0-9]{2}"/>
       <xs:minInclusive value="0"/>
       <xs:maxInclusive value="19"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
   </xs:complexType>
  </xs:element>

  <xs:element name="NCHAR">
   <xs:complexType>
    <xs:attribute name="Length" use="required">
     <xs:simpleType>
      <xs:restriction base="xs:positiveInteger">
       <xs:pattern value="[0-9]{3}"/>
       <xs:minInclusive value="2"/>
       <xs:maxInclusive value="100"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
   </xs:complexType>
  </xs:element>

  <xs:element name="OTHER">
   <xs:complexType>
    <xs:attribute name="Type" type="OTHERType" use="required"/>
   </xs:complexType>
  </xs:element>

 </xs:choice>
</xs:complexType>

<!-- Simple DataColumnTypeTypes -->
```

```
<xs:simpleType name="OTHERType">
 <xs:restriction base="xs:string">
  <xs:enumeration value="DATE"/>
  <xs:enumeration value="TEXT"/>
  <xs:enumeration value="BOOLEAN"/>
  <xs:enumeration value="BLOB"/>
  <xs:enumeration value="REAL"/>
  <xs:enumeration value="INTEGER"/>
 </xs:restriction>
</xs:simpleType>

<!-- SpatialColumnType -->
<xs:complexType name="SpatialColumnType">
 <xs:simpleContent>
  <xs:extension base="GeometryName">
   <xs:attribute name="Dimension" type="GeometryDimension" use="required"/>
   <xs:attribute name="Nullable" type="GeometryNullable" use="required"/>
  </xs:extension>
 </xs:simpleContent>
</xs:complexType>

<!-- Geometry Types -->
<xs:simpleType name="GeometryName">
 <xs:restriction base="xs:string">
  <xs:enumeration value="POINT"/>
  <xs:enumeration value="LINESTRING"/>
  <xs:enumeration value="POLYGON"/>
  <xs:enumeration value="MULTIPOINT"/>
  <xs:enumeration value="MULTILINESTRING"/>
  <xs:enumeration value="MULTIPOLYGON"/>
 </xs:restriction>
</xs:simpleType>

<!-- Geometric Dimensions -->
<xs:simpleType name="GeometryDimension">
 <xs:restriction base="xs:string">
  <xs:enumeration value="XY"/>
  <xs:enumeration value="XYZ"/>
  <xs:enumeration value="XYM"/>
  <xs:enumeration value="XYZM"/>
 </xs:restriction>
</xs:simpleType>

<!-- Geometry Nullable -->
<xs:simpleType name="GeometryNullable">
 <xs:restriction base="xs:nonNegativeInteger">
  <xs:pattern value="[0-1]"/>
  <xs:minInclusive value="0"/> <!-- Geometry Nullable -->
  <xs:maxInclusive value="1"/> <!-- Geometry not Nullable -->
 </xs:restriction>
</xs:simpleType>

<!--*****************Constraint-Datatypes******************-->

<xs:complexType name="ConstraintType">
 <xs:choice>
  <xs:element name="SimpleConstraint">
   <xs:complexType>
    <xs:simpleContent> <!-- Containing CHECK-Statement, if one exists -->
```

```
      <xs:extension base="xs:token">
       <xs:attribute name="Type" type="SimpleConstraintType" use="required"/>
      </xs:extension>
     </xs:simpleContent>
    </xs:complexType>
   </xs:element>
   <xs:element name="FOREIGNKEY">
    <xs:complexType>
     <xs:attribute name="RefTableName" type="NameType" use="required"/>
     <xs:attribute name="RefColumnName" type ="NameType" use="required"/>
     <xs:attribute name="DeleteAction" type="ForeignKeyActionType" use="required"/>
     <xs:attribute name="UpdateAction" type="ForeignKeyActionType" use="required"/>
    </xs:complexType>
   </xs:element>
  </xs:choice>
 </xs:complexType>

 <xs:simpleType name="SimpleConstraintType">
  <xs:restriction base="xs:string">
   <xs:enumeration value="UNIQUE"/>
   <xs:enumeration value="PRIMARY KEY"/>
   <xs:enumeration value="NOT NULL"/>
  </xs:restriction>
 </xs:simpleType>

 <xs:simpleType name="ForeignKeyActionType">
  <xs:restriction base="xs:string">
   <xs:enumeration value="NO ACTION"/>
   <xs:enumeration value="RESTRICT"/>
   <xs:enumeration value="SET NULL"/>
   <xs:enumeration value="CASCADE"/>
  </xs:restriction>
 </xs:simpleType>

 <!--*****************Global-Datatypes******************-->

 <!-- Type for all names. Only latin-letters with length 1 to 30 allowed. See van der
      Vlist 2002. -->
 <xs:simpleType name="NameType">
  <xs:restriction base="NameTypeType">
   <xs:pattern value="[\p{L}]*"/>
  </xs:restriction>
 </xs:simpleType>
 <xs:simpleType name="NameTypeType">
  <xs:restriction base="xs:NCName">
   <xs:minLength value="2"/>
   <xs:maxLength value="30"/>
   <xs:pattern value="[\p{IsBasicLatin}]*"/>
  </xs:restriction>
 </xs:simpleType>
</xs:schema>
```

Although the scheme's structure is intended to be self-explanatory for those familiar with $_{e}X_{tensible}M_{arkup}L_{anguage}$, several content-related project-specific aspects deserve mention:

The actual data-fields the schema describes resemble tables, columns and indices in a database-structure and contain, therefore, all the information necessary to build up that structure in a database-system.

The distinction between two different table-types takes account of the fact that a table containing a glossary is subject to different rules than a table for actual data-storage (see *2.3.1. Scheme*). The scheme, then, makes use of X-Path[35] elements, in order to not only enforce unique table- and column-names, but also to allow for referencing those names in relation with database-foreignkeys. The intention, therefore, is to incorporate into the scheme as much validity-checking as possible, which also holds true for defining data- or column-types quite strictly. The latter are construed from SQLite[36]/SpatiaLite[37] field types, but do not cover the entire spectrum of available options or functions[38]. Since being the major concern in this endeavour the support of SpatiaLite's geometry-types, build up the core issue. Additionally, the number of tables as well as columns per table is restricted to a reasonable degree.

Additionally entities and predicates of an ontology can be included, so as to allow for the schema's elements to be mapped to another model. The schema allows for ontology-entities to be assigned to tables and columns, whereby foreign-key-columns stand for the referenced table. Relations are expressed as predicates mostly assigned to columns, so that they are to be understood as a constellation of 'table-entity - column-predicate - column-entity' thereby forming a sort of sentence. In rare cases a table may carry a predicate instead of an entity, when its only purpose is representing a connection between two other tables in the manner of a many-to-many multiplicity.

The following chapter discussing the data-model actually applied in this project exemplifies the way the scheme can be implemented - thence illustrating the scheme's purpose.

2.2.2. Data-Model

Fortunately, the *Guidelines for Archaeological Measures in Austria* [BDA] equip this endeavour with a pool of data fields, most of which were directly incorporated into the model.

It deems reasonable to be geared by the forms provided by aforesaid guidelines, as they are not only published by the national agency for cultural heritage, but provide also the legal framework for archaeological measures in Austria and must, hence, be widely recognized.

In database-design several conceptual guidelines have been established as a means to lead towards consistent database-drafts. The chosen illustration (see Figure 3), for instance, widely follows conventions of entity-relationship-modelling. [see Barker 1999; Kleuker 2011] Aside from sheer graphical issues some core paradigms qualify sophisticated database-designs, as there are non-redundant data-storage, integrity as well as coordinating data-access and rights. [Kleuker 2011] While there are complete process models, that can be employed from scratch to ready database-model, the first steps in that procedure are jumped here, as the mentioned *Guidelines for Archaeological Measures in Austria* already bring an ordered set of entities and attributes. From basic entity-relationship-diagrams table-structures are usually deduced applying the well-established levels of normalization [see Kleuker 2011] consequently leading towards the database design displayed in Figure 3.

[35] see: https://www.w3.org/TR/xpath/ <last visited: 14.02.2017> and Vonhoegen 2013.
[36] see: https://www.sqlite.org/datatype3.html <last visited: 14.02.2017> and Owens 2006.
[37] see: http://www.gaia-gis.it/gaia-sins/spatialite-manual-2.3.1.html <last visited: 14.02.2017>.
[38] Consider for example AUTOINCREMENT or default-values for columns.

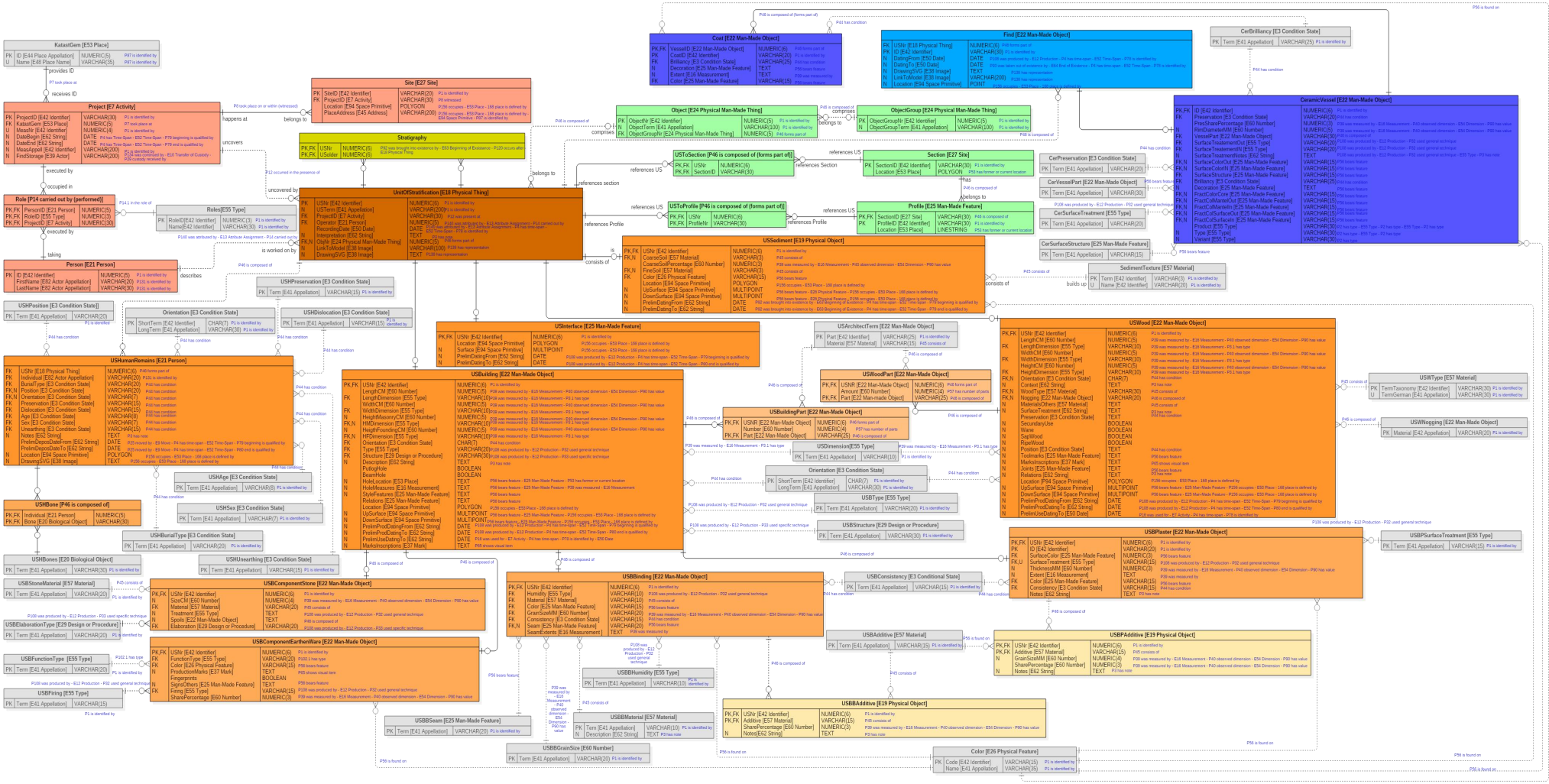


Figure 3

the data-model expressed as entity relationship model

(Blue coloured phrases represent entities and predicates of the ontology CIDOC CRM assigned in a way they seem most suitable. Also see chapter *2.2.3. Excursus: Ontologies*. Gray background-colour characterizes the table as glossary-table, while all other background-colours are just intended to convey a nice look.)

Deviations from the Guidelines

Geometries are, as a matter of course, directly stored in the database constituting the core advantage over the classic recording practice. All terms are in English, so as to promote unhindered access and understanding, since like said in the introductory chapter european-wide integration is definitely a desideratum. Hence, the challenge arises to employ suitable terms. Regarding human remains and burial practice the guidelines present several comments on tomb arrangements and fittings, that are better treated as features of their own (indicated on the accompanying forms anyway). Finds, on the other hand, are only concerned by the guidelines' forms in context with burials, so that the data model employs a separate finds-section providing - for demonstrating purposes only - tables for ceramic vessels. That section relates to the table of 'UnitOfStratification', in order to connect finds with their stratigraphic contexts (and not just with burials). Since omitted in the guidelines, the structure of the finds' tables are the authors product and supposed to be extended for various kinds of finds in the practical use. The very unspecific fields for dating must be considered a shortcoming of the guidelines' specifications, as there has to be a clear difference between dating the creation of a unit of stratification (e.g. constructing a building dated by marks) and the dating of the period of its use. Hence, there is also a clear semantic difference in terms of dating between different kinds of units of stratification, considering, for example, sediment, the deposition of which being the only dateable factor about it. In a similar manner other fields were added to the data model, where it appeared reasonable. Conversely, composition and additives of masonry-binding are combined into one field. The guidelines' forms for geodetic survey are left aside, because their information is not actually relevant for archaeological research. The same holds true for addresses of people involved. Multiple-choice-fields as well as those requiring a certain vocabulary representing a list of valid values are implemented as separate glossary-tables.

Entities and predicates of the ontology named CIDOC-CRM (see *2.2.3. Excursus: Ontologies*) were mapped to the corresponding model-items.

2.2.3. Excursus: Ontologies

When it comes to the main concern of this paper - namely integration and comparability of excavations' raw data -, the generated data-model cannot ignore common ontologies, the widest employed of which in cultural heritage clearly being CIDOC-CRM[39], an approved standard - as pointed out in the introductory chapter - equipped with the ability to name spatial entities. The conceptual reference model CIDOC-CRM, thereby, provides the common ground for unequally structured databases holding comparable content supposed to be integrated. The integration-process itself relies on mapping the specific model-items to the abstract ontology.

The benefit of the conceptual reference model rests with its inherent semantics exploiting resource description framework's[40] syntax together with principles of 'subject', 'predicate' and 'object'. Thanks to its abstractness it can unambiguously reside on top of the designed database scheme attaching its own abstract semantic entities to tables and columns whilst conferring additional implication and scope to the records. Not only can the data-model, thus, be easier joined with akin cultural-heritage-data - as it has already been mapped to CIDOC-CRM (a task practised also at *Research Infrastructure* ARIADNE) –, but the data model itself becomes more consistent having had to fit itself in with an approved ontology.

Efforts have already been made to develop special ontologies for archaeological excavation data, which at first glance appears beneficial. But before two of them are discussed closer, the very profound intention of ontologies has to be considered, namely

[39] Current version can be found at: http://83.212.168.219/CIDOC-CRM/Version/version-6.2 <last visited 14.11.2015>; http://83.212.168.219/ CIDOC-CRM/sites/default/files/cidoc_crm_version _6.2.pdf <last visited 14.11.2015>.
For information on the model see: http://www.cidoc-crm.org/ <last visited 14.11.2015>.

[40] see e.g. http://www.w3.org/RDF/ <last visited 20.11.2015>.

to integrate data across disciplines. Taking away abstractness by tailoring the ontology to a more specific domain means annihilating most of its core quality and degrading it almost to a simple data model.

This said, the CRM-EH developed by *English Heritage* deserves a note, as it maps archaeological processes and concepts of *English Heritage* to CIDOC-CRM.[41] Entities of the latter are, thereby, endowed with excavation-specific meaning, yet as long as the excavation-process is not sought to be conducted in the way of *English Heritage*, the CRM-EH - as not constituting an ontology by itself - hardly contributes to the data-model in this endeavour.

On the other hand, there is CIDOC-CRMarchaeo[42], a kind of refinement of the regular CIDOC-CRM in archaeological terms. This alleged enhancement introduces archaeological subclasses (to the regular classes), while at the same time mixing up not only different levels of abstractness, but also general with domain-specific classes - raising therewith the hazard of class-conflicts. Another critical point is the fact that the archaeological entities inherit from entity E53 Place (probably in order to make its predicates available) - a condition avoided in the regular conceptual reference model, where place is nearly always just a quality to events or things. Allowing archaeological entities to be descendants of E53 Place implies that a unit of stratification has a location and is a location at the same time, whereas its identity is apparently directly related to the location.

The conclusion drawn from the discussion above is that among aforesaid ontologies only CIDOC-CRM in its regular form (which was made into a standard) can be applied fruitfully, so as to contribute to the design of the data model. In that way the ontology bridges the semantic-gap between disciplines, whereas intending to integrate data exclusively within the domain of archaeological excavation an approach of either providing a truly specified and conclusive ontology or a data model should be taken. The latter is sought for by this paper.

[41] see: http://hypermedia.research.southwales.ac.uk/resources/crm/ <last visited 14.11.2015> and: http://hypermedia.research.southwales.ac.uk/media/files/documents/2010-05-05/Arch_Ontological_ModellingV4.pdf <last visited 14.11.2015>.

[42] see: http://www.ics.forth.gr/index_main.php?l=e&search type=st_p&c=711&i= <last visited 14.11.2015> and http://www.cidoc-crm.org/docs/cidoc_crm_sig/CRMarchaeo_v1.3.pdf <last visited 14.11.2015>.

2.3. GLOSSARIES

Where the *Guidelines for Archaeological Measures in Austria* do not provide the terminology to fill the (multiple-choice-) fields with and a systematic nomenclature is clearly desired, vocabularies are sought to be employed, so as to purvey documented and recognised terminologies. Since glossaries are subject to similar requirements as the data-model, the same technology for encoding is applied, namely $_{e}X_{tensible}M_{arkup}L_{anguage}$ (*see 2.2. Data-Model*), in order to structure glossary-documents. The following scheme, therefore, describes the valid composition of such documents, so that a measure of standardisation is provided. The schema defines 'Table' as root-element and a sequence of rows as its child-elements, whereupon a 'Row' is made up of exactly one 'PrimaryKeyColumn'-element and zero or more 'Column'-elements. The table as well as each of the columns possess an attribute called 'Name' making a valid glossary-document reflect rows and columns in a table. Although not illustrated by the scheme, there are certain conceptual limitations to glossaries, as there are:

While in a database a table may feature composite primary-keys, that concept hardly deems reasonable for a glossary, as the term the glossary is supposed to provide takes that part. Likewise, all columns must only contain alphanumeric values by reason that the glossary-terms will be engaged as foreign-keys within another table, whereas the remaining columns are solely meant to describe the terms in more detail. Binary data, for instance, would have to be encoded in hexadecimal notation, in order to be stored in XML-documents, and would, therefore, extend the documents unnecessarily. Additionally, glossaries are not expected to employ foreign-keys themselves, as they only provide vocabulary. They are ending points, respectively top-elements, within any data-model.

2.3.1. Scheme

```
<?xml version="1.0" encoding="UTF-8"?>

<!--  Author: Mag. Thomas Winklehner
  Purpose/Description: Schema-Document for Glossaries
  Date: 25.04.2016
  Warning: DO NOT MODIFY THIS DOCUMENT
-->
<!-- using XSD 1.0 -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.thomaswinklehner.at/Glossary"
  xmlns="http://www.thomaswinklehner.at/Glossary"
  xmlns:tw="http://www.thomaswinklehner.at/Glossary"
  elementFormDefault="qualified">
 <xs:element name="Table">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="Row" minOccurs="0" maxOccurs="unbounded">
     <xs:complexType>
      <xs:sequence>
       <xs:element name="PrimaryKeyColumn" minOccurs="1" maxOccurs="1">
        <xs:complexType>
         <xs:simpleContent>
          <xs:extension base="xs:string">
           <xs:attribute name="Name" use="required" type="xs:string" />
          </xs:extension>
         </xs:simpleContent>
        </xs:complexType>
       </xs:element>
       <xs:element name="Column" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
         <xs:simpleContent>
          <xs:extension base="xs:string">
           <xs:attribute name="Name" use="required" type="xs:string" />
          </xs:extension>
         </xs:simpleContent>
        </xs:complexType>
       </xs:element>
      </xs:sequence>
     </xs:complexType>

     <xs:unique name="ColumnUniqueness">
      <xs:selector xpath="tw:Column"/>
      <xs:field xpath="@Name"/>
     </xs:unique>

    </xs:element>
   </xs:sequence>
  <xs:attribute name="Name" use="required" type="xs:string"/>
  </xs:complexType>

  <xs:unique name="PrimaryKey">
   <xs:selector xpath="tw:Row/tw:PrimaryKeyColumn"/>
   <xs:field xpath="."/>
  </xs:unique>

 </xs:element>
</xs:schema>
```

2.4. THE DESKTOP-APPLICATION

As with creating the data-model the first concern of assembling the desktop-application is rooted in settling for a suitable technology to employ - as exemplified by the subsequent section.

2.4.1. Programming Language and Framework

In order not to repeat the reasons presented above (chapter *1.2.3. Conceptual Solution*) in favour of C/C++ as the most suitable programming language for this project, only the following quotation shall be cited, so as to stand for a brief paraphrase on the language:

"C++ is an object-oriented, low-level ANSI and ISO standard programming language. As a low-level language similar to and compatible with its predecessor C, C++ can generate very efficient, very fast programs. As an object-oriented language, C++ has the power and extensibility to write large-scale programs. C++ is one of the most popular programming languages for all types of programs. Most of the programs you use on your PC every day are written in C++. C++ has been certified as a 99.9 percent pure standard. This makes it a portable language. There is a C++ compiler for every major operating system, and they all support the same C++ language. (Some operating systems support extensions to the basic language, but all support the C++ core.)"[Davis 2004][43]

Having determined the preferred programming language the next methodological building block towards setting up the desktop-application is finding a framework for the GraphicalUserInterface. As to not being forced to produce several source-code packages - one per operating system (OS) and compiler -, a framework named Qt[44] will be referenced in this endeavour.

"Qt is a comprehensive C++ framework for developing cross-platform GUI applications using a 'write once, compile anywhere' approach. Qt lets programmers use a single source tree for applications that will run on Windows [...], Mac OSX, Linux, Solaris, HP-UX, and many other versions of Unix with X11. The Qt libraries and tools are also part of Qtopia Core, a product that provides its own window system on top of embedded Linux." [Blanchette –Summerfield 2006]

A useful feature of Qt not mentioned yet is its signal-slot-mechanism, in order to trigger functions on certain events.[45] Qt comes with diverse licensing options [https://www.qt.io/download/ <last visited: 15.02.2017>], one of which is the so called GNU Lesser General Public License (LGPL), that can be employed without charge under certain obligations. [https://www.qt.io/qt-licensing-terms/ <last visited: 15.02.2017>; https://www.qt.io/faq/#_Toc_3_7 <last visited: 15.02.2017>] Although the latter are neglected in the course of this paper, as the resulting software is neither commercially sold nor published in any way, LGPL stands for the option aspired for - accordingly bringing about the duty of dynamically binding the desktop application to the Qt-libraries - a relevant procedure in terms of licensing[46]. A tool commonly supplied with the Qt-packages is Qt Creator[47], an IntegratedDevelopmentEnvironment, that by default has dynamic linking enabled and was not only by that reason used for this project. There is, of course, a documentation on Qt's capabilities[48] accessible online, with regard of which one may bear in mind Qt 5.7 being the current version during this application's development phase.

2.4.2. GUI-Overview and the Core Principle

The subsequent page gives a visual overview of the application's GraphicalUserInterface, in order to emit a general impression of the program as well as a first graphical depiction of the major functional groups. Thereafter a few central characteristics are to be bespoken.

[43] For more information on the topic see: Deitel – Deitel 2012, Horton 2014, Josuttis 2012 or Stroustrup 2013.
[44] see: https://www.qt.io/ <last visited: 15.02.2017>.
[45] see: http://doc.qt.io/qt-5/signalsandslots.html <last visited: 19.02.2017>.
[46] see: http://www.gnu.de/documents/lgpl-3.0.de/html <last visited: 16.02.2017> or https://www.gnu.org/licenses/lgpl-3.0.de.html <last visited: 16.02.2017>.
[47] see: https://www.qt.io/ide/ <last visited: 16.02.2017>.
[48] http://doc.qt.io/qt-5/ <last visited: 16.02.2017>.

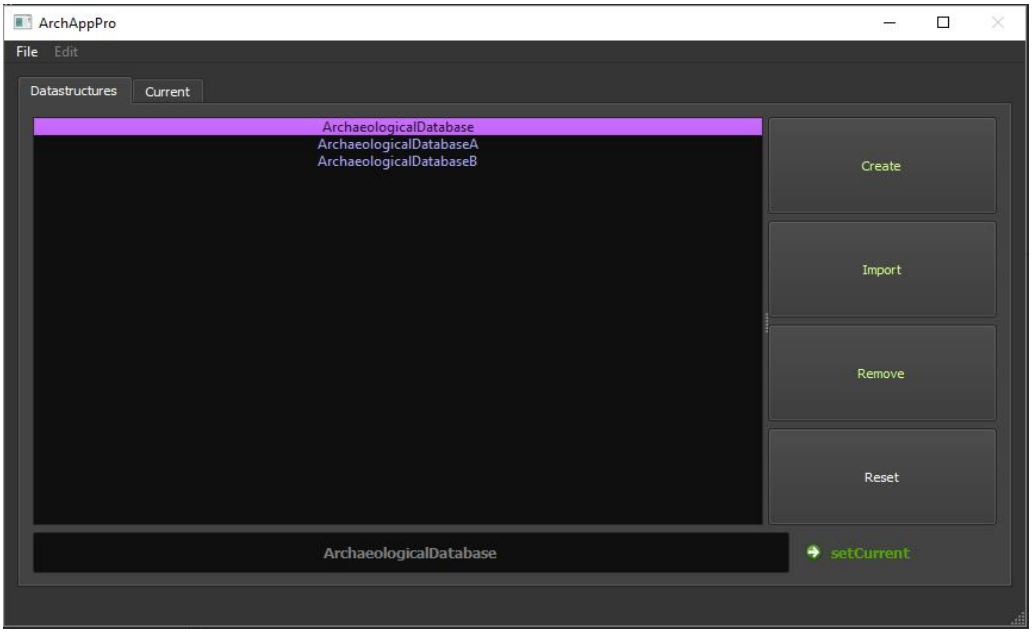


Figure 4

GUI after application started with basic data-model-/data-structure-functionality enabled

The GUI's appearance is kept dark, so as to reduce tiring of the user's eyes. The applied settings were found at https://gist.github.com/skyrpex/5547015 [last visited: 07.05.2016]

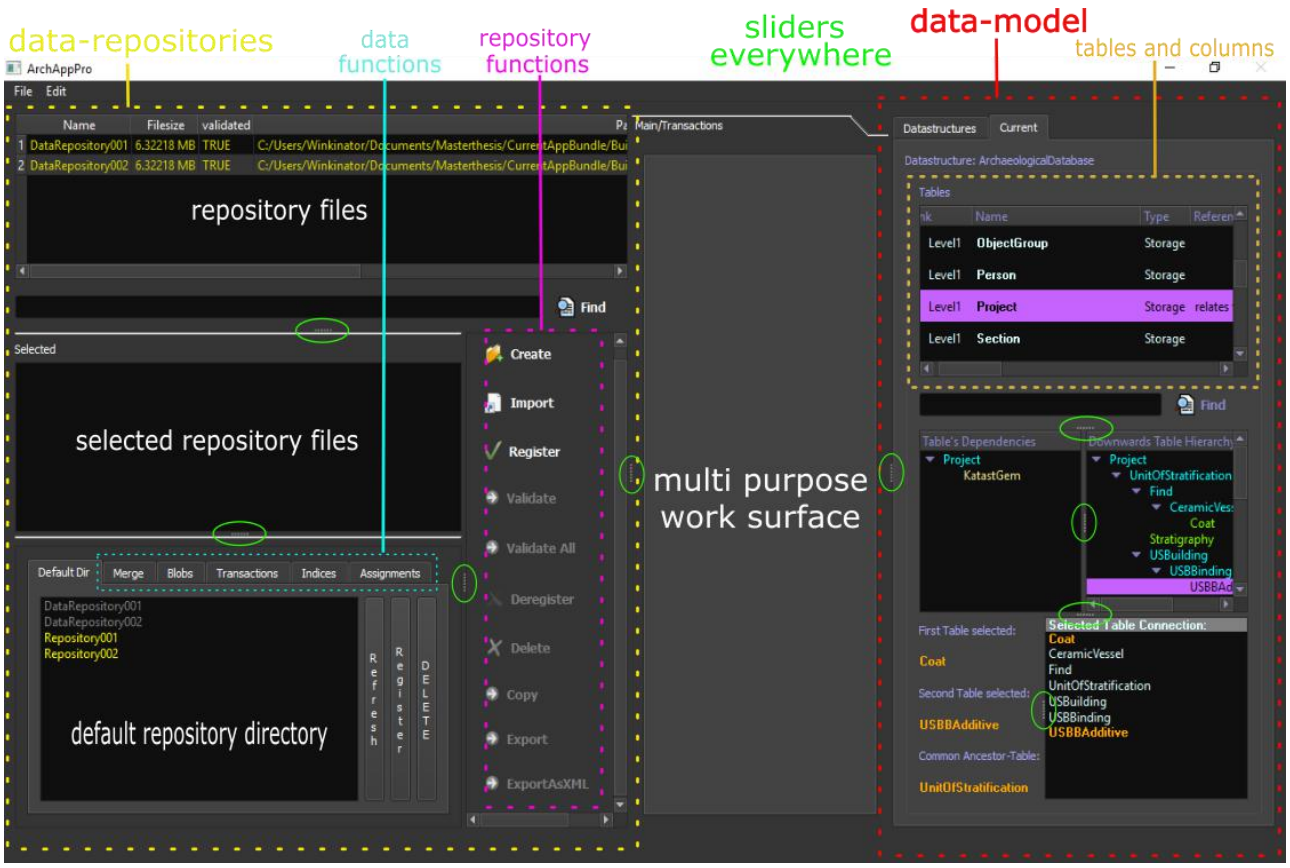


Figure 5

GUI after data-model selected (normal state)

First of all, Figure 4 and Figure 5 not only represent the GUI's two major kinds of appearance, but also illustrate the essential practice of selecting a data-model - henceforth also called data-structure - (Figure 4), before advanced functionality can be used (Figure 5). It is, hence, not until the application enters its (default) tripartite appearance of Figure 5, before most of its functionally grouped by capabilities become even apparent due to the latter calling for a valid data-model and the GUI generally enabling only employable widgets by reason of user friendliness. A basic feature, thence, is preventing the change of data-models as long as certain tasks are active. Additionally, as shown by Figure 5, the GUI makes excessive use of window-resizing-sliders, in order to empower the user to enlarge any major panel up to nearly the size of the application's main window.

In order to strive the visual functional components of Figure 5, it is to be stated that the pane to the right - including the widgets of Figure 4 by the way - covers all data-model-related functionality, while the pane at the left handles all data-repository-related capacity as well as all sorts of database-transactions. In between there lies the main work surface deployed for displaying forms, query-results or a simple map, whereas at the top the edit-menu provides certain helper-functionality.

Core Principle

For the sake of a positive user-experience the application must pursue a policy of self-consistency and validity - meaning that the program has to protect itself against an inconsistent state or crashing due to an error. Roughly at least half of the application's logic is engaged in the task of state-checking and self-healing trying, however, not to slow down the processes noticeably. The state-checking and self-healing begins at the start-up of the application, when external dependencies are verified, and includes the attempt of restoring important files like the data-model-scheme (see *2.2.1. Scheme*) and de-registering invalid data-models or database-files. In a nutshell, it is commonly the external files and directories, that might suffer from corruption (probably due to user-interaction) and have to be repaired or replaced by the program.

2.4.3. External Dependencies

As hinted by chapter *1.2.3. Conceptual Solution* and Figure 2 the desktop-application not only depends on certain code-libraries, but also on external database-files and documents. Although dynamic libraries do not need to be stored with the executable, for the sake of simplicity, however, all files indispensable for the application to work are bundled into a single directory as illustrated by Figure 6.

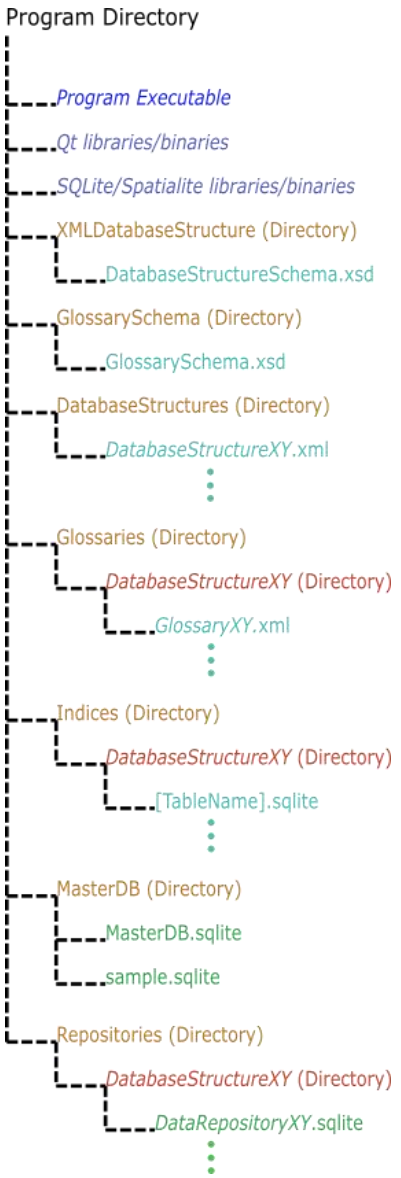


Figure 6

schema of nested directories and files

Beside the already mentioned Qt-libraries/binaries as well as the schemes for glossaries and data-models (data-structures) Figure 6 exposes the existence of a directory for the data-model documents as well as directories for glossary-documents, database-files utilised as indices (see chapter *2.4.5.12. External Indices*) and repositories, each subdivided by directories named after available data-models. The folder containing the so called master-database also holds a database-file named sample.sqlite, which is needed as prototype for newly created repositories. Like Qt SpatiaLite (see chapter *2.4.4.3. SQLite and SpatiaLite*) depends on certain libraries/binaries preferably stored with the executable, even though the way linked program-libraries are looked up is determined by the operating system.

2.4.4. Main Internal Structure

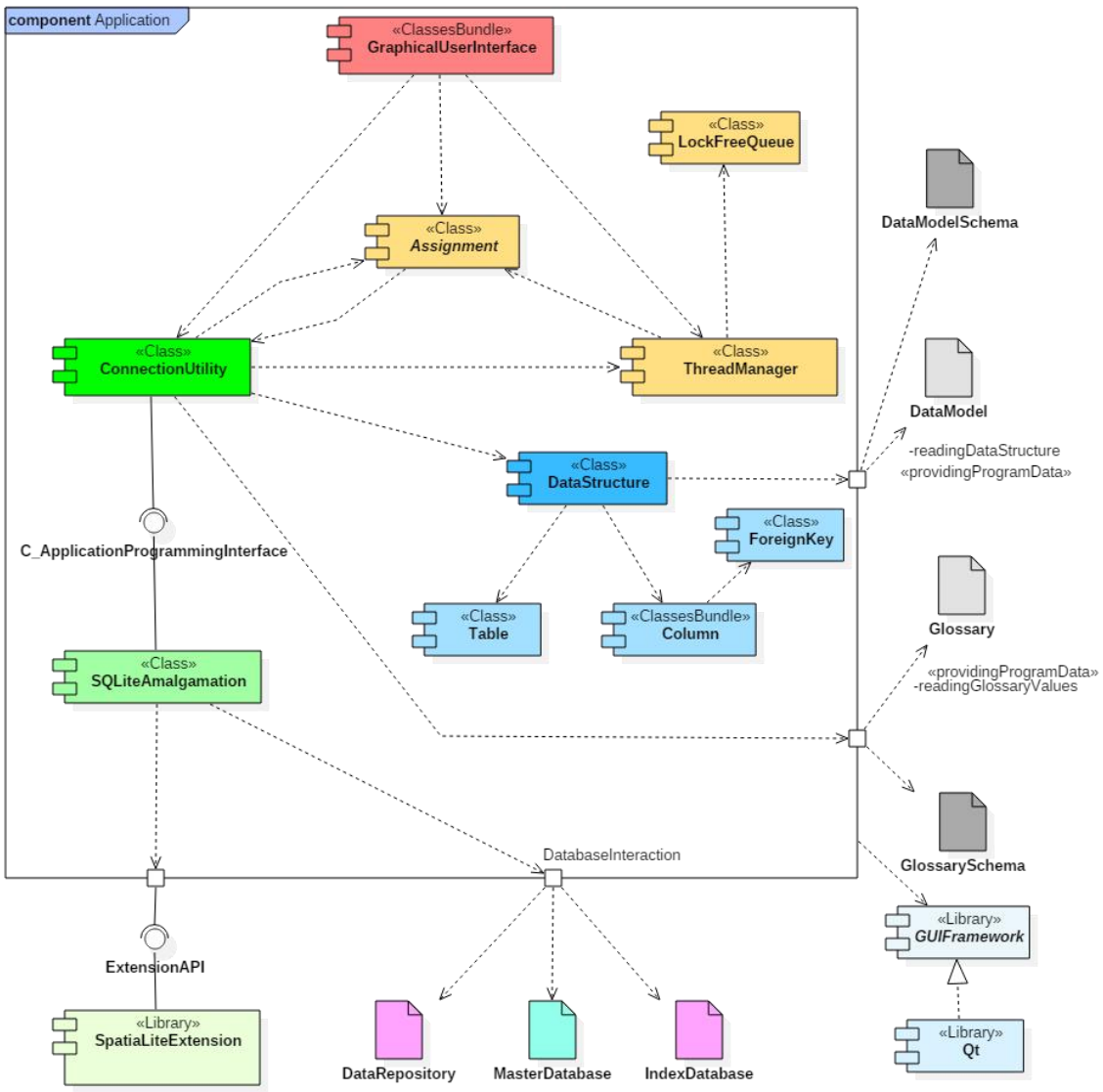


Figure 7

UML-compliant diagram of the desktop-application

Figure 7 visualises the simplified main internal structure of the desktop application - schematically composed of classes or class-bundles, the latter representing several classes combined into a single component. The illustration cannot be exhaustive, since it only depicts the components created by the author leaving aside all the referenced classes of the C++ Standard Template

Library[49] and the Qt-libraries. Figure 7, however, suffices to discern the rough overall functionality featuring a GraphicalUserInterface at the top-end being supplied advanced capabilities chiefly by two other classes, the ConnectionUtility and the ThreadManager themselves referencing the fourth class called Assignment. The reason for that kind of heavily interconnected arrangement is rooted in the attempt to implement concurrency put into effect mostly by the ThreadManager-class. If the latter cares about, how things are carried out on the operating system, it is usually the ConnectionUtility specifying what has to be done. The Assignment-class, thereby, serves as vehicle between the instances forming the transport-box enclosing the function intended to be scheduled for execution on a separate thread by the ThreadManager. In doing so, the GUI constitutes commonly both, starting point as well as the projection area for the executed function's results. The ConnectionUtility-class for its part relies on the one hand on SQLite for basic database-tasks, on the other hand on a couple of classes representing the data-model.

Even though it generally appears good practice to source out expensive tasks to worker-threads, in order not to have the GraphicalUserInterface blocked, the key concept built into the application's structure is its ambition to exploit the hardware's resources in terms of concurrency - a.k.a. multithreading - at most. As the program will have to handle several databases at once, parallel processing not only comes in handy, but is desperately in need. The theory behind it additionally presumes that the workload will be somewhat related to hardware capacities, as one would hardly run large-scale projects on a poorly performing machine. The requirement of scaling well with the hardware is, hence, fulfilled, too.

Although the extent of this paper does not suffice to illustrate all the source code - not even the content of the header files -, the following sections discuss the various components in more detail mainly utilizing conceptual graphics rather than long episodes of source code.

[49] see: Josuttis 2012.

2.4.4.1. ThreadManager, Assignment and LockFreeQueue

Beside referencing C++- and Qt-libraries the application consigns three of its components, namely ThreadManager, Assignment and LockFreeQueue to bring concurrency to life. Like revealed by Williams [2012] - the source of information on concurrency concerning C++ in regard of this paper - creating a thread and assigning it a task for execution is not a difficult venture, managing access to shared variables and resources, however, is the key challenge. While in this regard there are locks, so called mutexes, to serialize sharing, more sophisticated methods work lock-free trusting in the application's design or atomic variables. [Williams 2012] The desktop-application utilizes both approaches and although there is both, concurrency support in Qt[50] as well as in native C++, only the latter is applied here. In terms of native C++-concurrency and Qt's signal-slot-mechanism, a certain behaviour is to be noticed with Qt being capable of receiving signals only on the GraphicalUserInterface's thread - apparently due to its event loop. The mechanism accomplished by the three aforementioned components roughly works in such a way that the task to be done is specified in an Assignment-object and handed over to the ThreadManager-object, which schedules the task for execution by storing and retrieving it in a lock-free queue.

Assignments

Figure 8 illustrates the linage of classes within the Assignments-component, that comprises the abstract class AbstractAssignment in its role as interface together with its descendants. Although in Figure 8 class-members are mostly left aside for reasons of plainness, the principle called polymorphism[51] employed for fabricating a kind of interface clearly becomes manifest in the abstract function named execute, which triggers the processing of the actual task.

[50] see: http://doc.qt.io/qt-5/qtconcurrent-index.html <last visited: 19.02.2017>.

[51] see for example: Deitel – Deitel 2012.

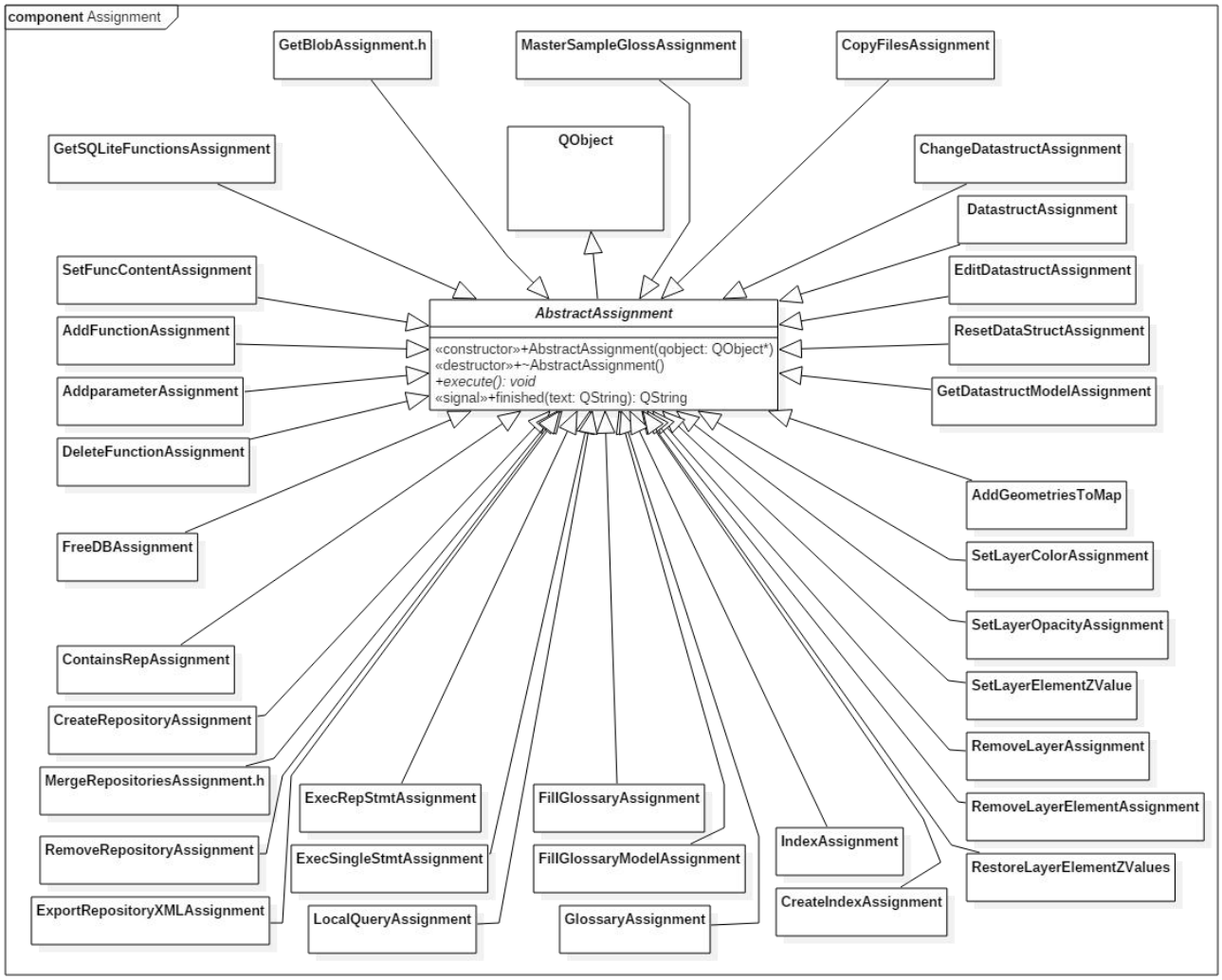


Figure 8

class diagram of assignments displaying only members of AbstractAssignment

---

The multitude of AbstractAssignment's child-classes is well-founded in the insight that diverse tasks require different sets of parameters accompanying the various functions to execute. Inheritance from QObject is due to assignments sending out a certain signal indicating completion. Additionally, some assignments apply more complicated mechanisms, in order to coordinate their behaviour as a group. Assignments neither have to be threadsafe themselves nor require a copy-constructor, since they are instantiated as pointers.

LockFreeQueue

The LockFreeQueue emulating a similar class in Williams [2012] is the very heart of the ThreadManager and facilitates queuing assignments later picked up for execution by free threads. This queue-class is based on the principle of a linked list and is built threadsafe without employing locks. The threadsafe behaviour is the result of code design, atomic variables representing head and tail of the queue and a dummy-node. See the following Diagrams:

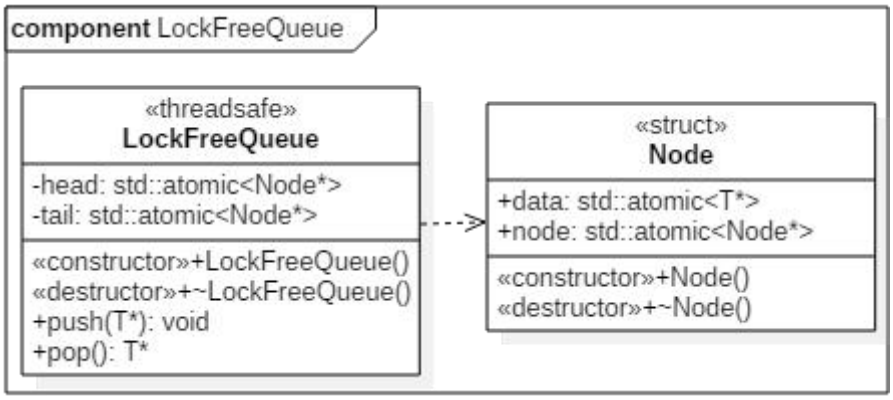


Figure 9

class diagram of LockFreeQueue

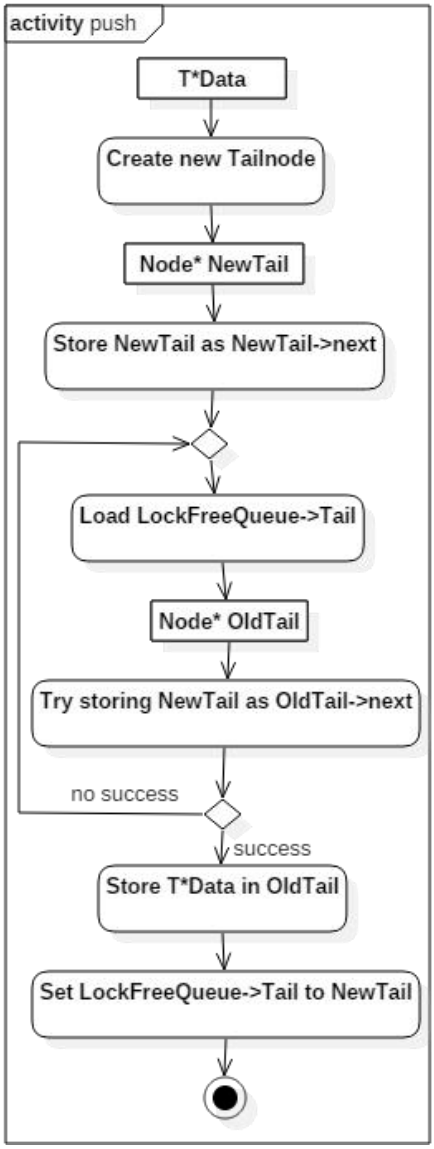


Figure 10

activity diagram of push-method

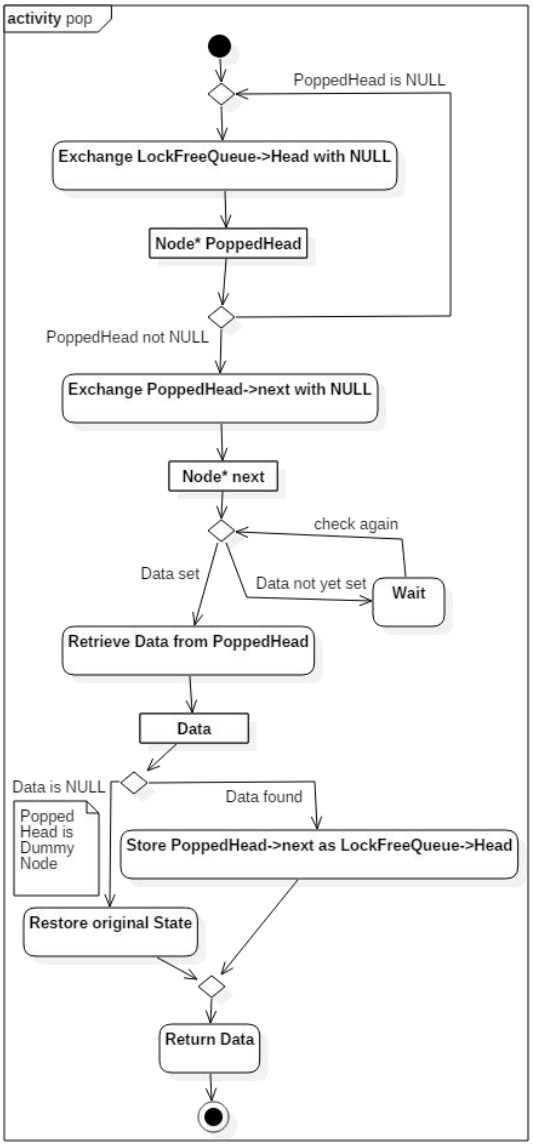


Figure 11

activity diagram of pop-method

ThreadManager

First thing to notice is the ThreadManager-class being instantiated statically and only once, for any other number of instances would corrupt the whole undertaking due to the necessity of the component holding the application-wide monopoly on threads. The main point about its internal processes, however, is that the expected workload will most certainly vary heavily and neither is the component to hold a large amount of system-resources in times of leisure, nor is it to hamper progress under high load. Additionally, any thread delivering assignments is not to be hindered in depositing the tasks due to a busy assignments-queue, but to be freed again as quick as possible.

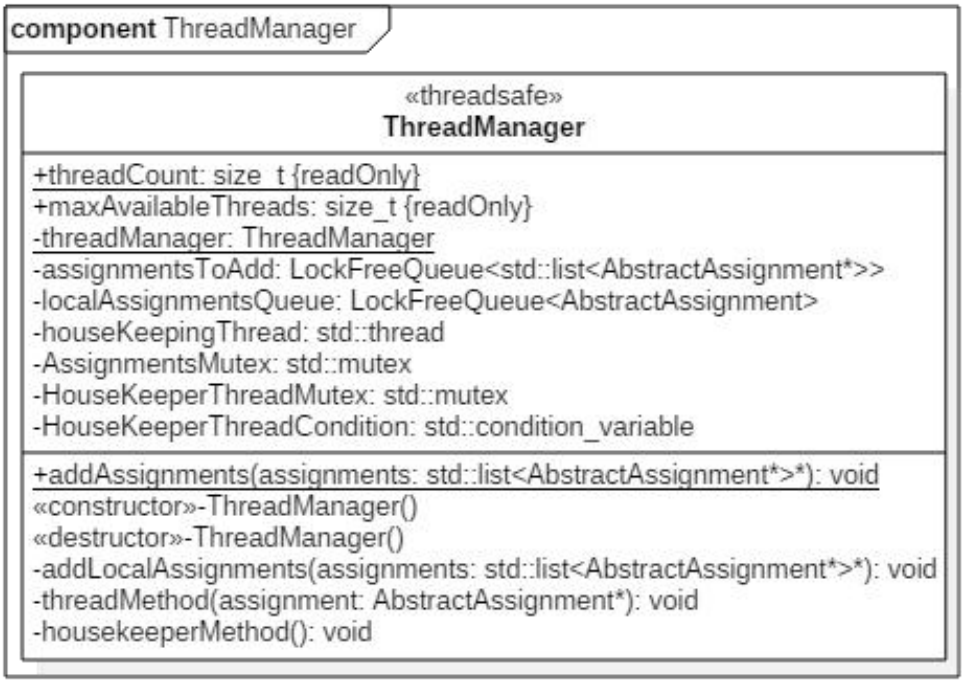


Figure 12

class diagram of ThreadManager

So, in a nutshell: The class holds a private static instance of itself and provides only one static public function for adding assignments. The latter are deposited as a single pointer, so that the delivering thread is freed immediately and does not have to accomplish pushing the single assignments onto the queue itself. As soon as an assignments-bundle arrived, the housekeeperMethod being executed on a separate thread is triggered - if not already running - adding the assignments to the queue and scheduling worker-threads for popping assignments off the queue. While the housekeeperMethod releases all its acquired resources and goes to sleep, if it has not been triggered again yet, the worker-threads not only keep picking up assignments and working them off by themselves (calling assignment's function execute()), but also kick-off further worker-threads, as long as there are assignments and threads left. After all work is done, all threads are freed, resources released and pointers deleted. The TreadManager, thereby, does not make any effort of synchronising variables referenced directly or indirectly by assignments.

Although the portrayed process may appear straightforward, it takes two mutexes and a condition-variable, in order to simply coordinate adding assignments, while additionally a LockFreeQueue, careful code-design, an atomic-variable and a lock-free assignments-dispenser-mechanism are in use. Prime hazards to avoid are dead-locks, thread-blocking and memory-leaks.

2.4.4.2. Data-Structure/Data-Model Classes

The DataStructure-class together with its related components represents the embodiment of the data-model and is, therefore, the interface between advanced programmatic functions and the data-model documents. This bundle of classes operates on five major fields of responsibility, namely reading and writing data-model documents, validating the data-structure, supplying all data-model-related information to the ConnectionUtility-component, building SQL-Create-statements and setting up a table-order based on foreign-keys. In order to fulfill its tasks, a DataStructure-object holds a container of Table-objects, which in turn each contains a map of AbstractColumn-objects representing different instances of the non-abstract column-classes:

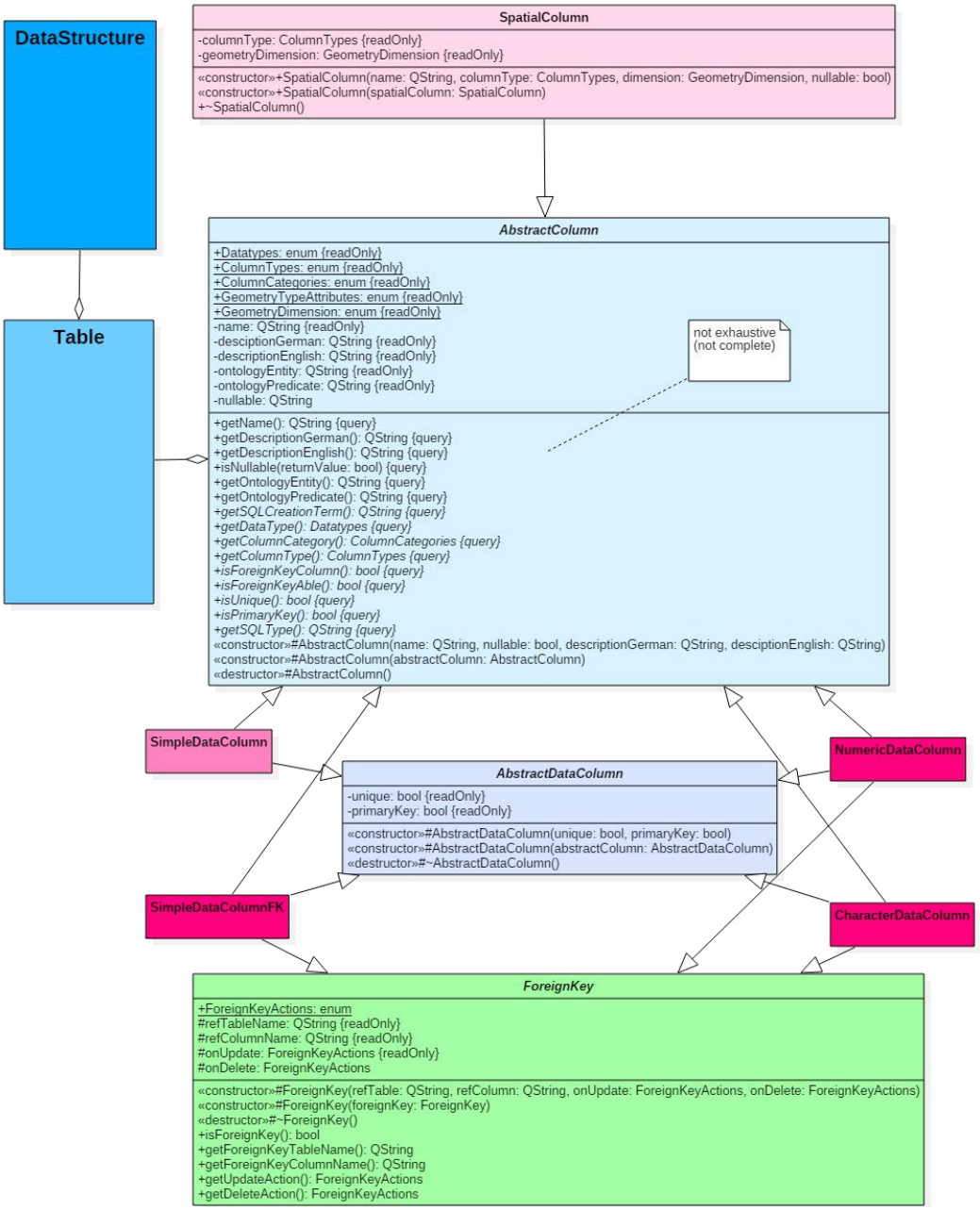


Figure 13

class-diagram of Table- and Column-classes

Figure 13 intends to convey a glimpse of the pretty straightforward character of the data-model-component, for a detailed stochastic treatise of the Table- or the DataStructure-class could easily double the extent of this paper without, however, contributing essentially to the general understanding of the application. The basic purpose of this component rests in retrieving, storing and dispensing information on the data-structure.

In order to review the important features, though, beginning with the most elementary, the mechanism of reading and writing $_{e}X_{tensible}M_{arkup}L_{anguage}$-files, is to be elucidated by remarking that the utilised Qt classes rely on the so called $D_{ata}O_{bject}M_{odel}$-tree[52]. This means that $_{e}X_{tensible}M_{arkup}L_{anguage}$-elements and -attributes are not read or written line by line, but in a single event constituting a full tree of nodes. In contrast to reading serially the downside of this mechanism is a somewhat reduced capacity in terms of document-size, while providing an easy-to-use interface to $_{e}X_{tensible}M_{arkup}L_{anguage}$-documents.

As in this context writing a valid data-model to file barely stands for a major obstacle, validating such structure after having read a data-model from file is a central concern. The first level of validation involves verifying the $_{e}X_{tensible}M_{arkup}L_{anguage}$-document with the XSD-schema by applying a QXmlSchemaValidator[53]. Consequently, not only do Table-objects control column-types, naming-conventions and -uniqueness, but the DataStructure-object conducts similar measures in regard of tables.

Additionally, the constraints enumerated in chapter *2.3. Glossaries* have to be enforced by the data-structure. Besides, every table must have at least one primary-key-column, which must not be spatial, and needs to be related to at least one other table via foreign-key. On the other hand, foreign-keys must not lead in circles, for the model actually has to form some sort of tree. The specified indices have to refer to columns actually existing, while indices for foreign-key-columns as well as spatial-columns are created by the program automatically. If the component is unable to establish a table order from foreign-key-relations (the less a table references other tables, the closer it is to the root), the validation is to fail.

Another major responsibility of the component is generating SQL-Create-statements for setting up the data-structure in database-files. The statements are, thereby, prepared as a single transaction, so as to succeed or fail completely not leaving behind an inconsistent state. By the way, a similar focus on consistency holds true for the component itself, in that it involves large parts of the source-code with file- and consistency-checking, so that only a valid data-model gets represented by the component.

2.4.4.3. SQLite and SpatiaLite

SQLite[54], the database-system selected out of reasons already laid out (see chapter *1.2.3. Conceptual Solution*), can be used completely freely, as stated by the following quotation:

“All of the code and documentation in SQLite has been dedicated to the public domain by the authors. All code authors, and representatives of the companies they work for, have signed affidavits dedicating their contributions to the public domain and originals of those signed affidavits are stored in a firesafe at the main offices of Hwaci. Anyone is free to copy, modify, publish, use, compile, sell, or distribute the original SQLite code, either in source code form or as a compiled binary, for any purpose, commercial or non-commercial, and by any means.[...]” [https://www.sqlite.org/copyright.html <last visited: 21.02.2017>]

Another advantage of SQLite over other database-systems is its source code amalgamation[55] being ready for inclusion directly into the application’s source code, so that the interaction with SQLite follows the rules of C++. Since SQLite permits several compile-time options[56], some of which impact the result heavily, corresponding parameters have to be set in the development environment or -

[52] see: http://doc.qt.io/qt-5/xml-dom-tml.html <last visited: 21.02.2017>.
[53] see: http://doc.qt.io/qt-5/QXmlSchemaValidator.html <last visited: 21.02.2017>.
[54] see: https://www.sqlite.org/<last visited: 21.02.2017>.
[55] see: https://www.sqlite.org/download.html <last visited: 21.02.2017>.
[56] https://www.sqlite.org/compile.html <last visited: 21.02.2017>.

as practised here - the sqlite-source-code is compiled to object-code applying aforesaid options in MinGW[57], before the object-code-file is included in the application's code. The compile-command reads as follows:

```
gcc -c sqlite/sqlite3.c -o sqlite/sqlite3.o -Os
-DSQLITE_THREADSAFE=2
-DSQLITE_LIKE_DOESNT_MATCH_BLOBS
-DSQLITE_MAX_EXPR_DEPTH=0
-DSQLITE_OMIT_DEPRECATED
-DSQLITE_DEFAULT_AUTOMATIC_INDEX=1
-DSQLITE_DEFAULT_FILE_FORMAT=4
-DSQLITE_DEFAULT_FOREIGN_KEYS=1
-DSQLITE_DEFAULT_LOCKING_MODE=0
-DSQLITE_DEFAULT_SYNCHRONOUS=3
-DSQLITE_DEFAULT_WAL_SYNCHRONOUS=3
-DSQLITE_DEFAULT_WORKER_THREADS=30
-DSQLITE_MAX_WORKER_THREADS=50
-DSQLITE_MAX_ATTACHED=100
-DSQLITE_CASE_SENSITIVE_LIKE
-DSQLITE_TEMP_STORE=1
-DSQLITE_ENABLE_COLUMN_METADATA
-DSQLITE_ENABLE_FTS3
-DSQLITE_ENABLE_FTS3_PARENTHESIS
-DSQLITE_ENABLE_FTS3_TOKENIZER
-DSQLITE_ENABLE_FTS4
-DSQLITE_ENABLE_FTS5
-DSQLITE_ENABLE_ICU
-DSQLITE_ENABLE_JSON1
-DSQLITE_ENABLE_LOCKING_STYLE
-DSQLITE_ENABLE_MEMORY_MANAGEMENT
-DSQLITE_ENABLE_PREUPDATE_HOOK
-DSQLITE_ENABLE_RBU
-DSQLITE_ENABLE_RTREE
-DSQLITE_ENABLE_SESSION
-DSQLITE_ENABLE_UNKNOWN_SQL_FUNCTION
-DSQLITE_ENABLE_UNLOCK_NOTIFY
```

Although among the enumerated parameters several only serve as forethought in terms of extending the application in future, probably the most important of the other options are SQLITE_THREADSAFE and SQLITE_DEFAULT_SYNCHRONOUS, because they facilitate concurrent access to single database-files[58]. Even though concurrency is additionally managed in the application's logic, the databases will be accessed by server-side scripts, too, thereby bypassing the program's logic.

In order to accomplish foreign-key-enforcement as well as spatial indices, SQLITE_DEFAULT_FOREIGN_KEYS = 1 and SQLITE_ENABLE_RTREE are set. The multitude of available options gives an impression of SQLite's capabilities, only a subset of which is actually employed in the desktop application.

SpatiaLite, on the other hand, is a spatial extension to the database-system and subject to slightly different open-source licensing terms similar to Qt.[59] SpatiaLite has to be compiled into operating-system-specific libraries[60], in order to be referenced by SQLite. Since the application is programmed on a windows operating-system, this endeavour luckily can fall back to pre-built windows binaries[61].

SQLite's mechanism of loading an extension is - from the user's point of view - based on communicating the file-path of the extension to the program, which may be achieved via C-$A_{pplication}P_{rogramming}I_{nterface}$-function.[62] Since due to licensing issues dynamic binding via shared libraries is to prefer over static linking anyway, a certain variation of the dynamic process is available (and therefore utilised here), making SpatiaLite a purely loadable module connected by SQL-statement.[63] After loading succeeded, spatial meta-data has to be initialised by the SQL-command "SELECT InitSpatialMetaData()" [http://www.gaia-gis.it/gaia-sins/spatialite-cookbook/html/metadata.html <last visited: 23.02.2017>], which can take up to several minutes to finish.

[57] see: www.mingw.org/ <last visited: 21.02.2017> or Gough – Stallman 2005.
[58] see: https://www.sqlite.org/threadsafe.html <last visited: 21.02.2017> and https://www.sqlite.org/pragma.html#pragma_synchronous <last visited: 21.02.2017>.
[59] see: https://www.gaia-gis.it/fossil/libspatialite/index <last visited: 21.02.2017>.
[60] see: www.gaia-gis.it/gaia-sins/sqlite-doxy-4.3.0/index.html <last visited: 21.02.2017> and www.gaia-gis.it/gaia-sins/ <last visited: 21.02.2017>.
[61] Available at: www.gaia-gis.it/gaia-sins/ <last visited: 21.02.2017>.
[62] https://www.sqlite.org/loadext.html <last visited: 23.02.2017>.
[63] https://www.gaia-gis.it/fossil/libspatialite/wiki?name=mod_spatialite <last visited: 23.02.2017>.

Although SQLite's C-A$_{pplication}$P$_{rogramming}$I$_{nterface}$ comprises a whole bundle of functionality[64], the functions actually employed in the desktop-application center mostly upon establishing a database-connection, setting pragmas, loading the spatial extension and executing database-transactions. The following activity-diagram is to illustrate the steps to take towards a valid database-connection.

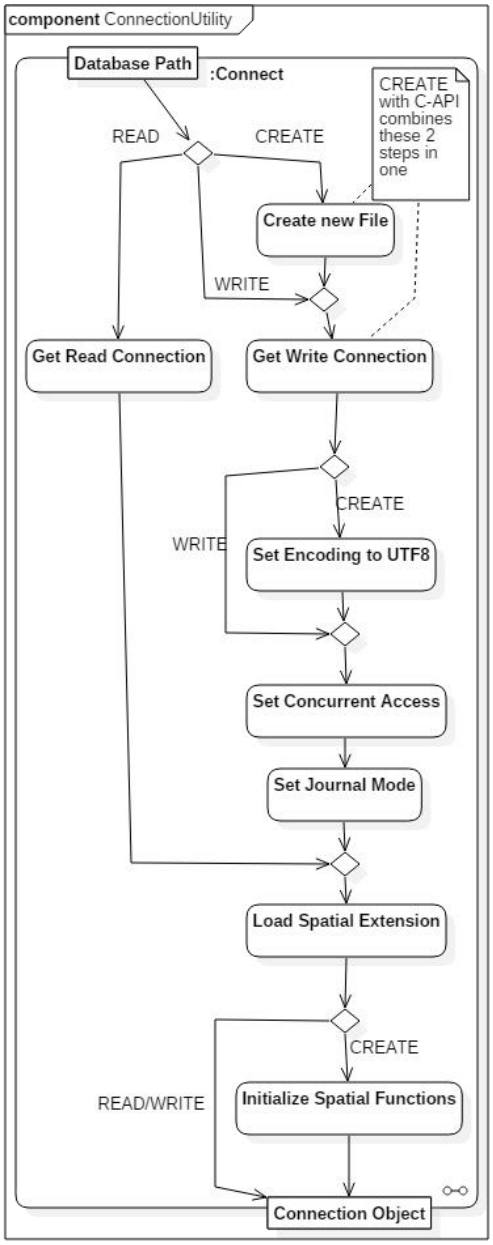


Figure 14

activity diagram of connecting to database

Concerning SQLite's journal mode (see Figure 14) - a feature facilitating rollback in transactions - the most sophisticated mechanism is W$_{rite}$A$_{head}$L$_{ogging}$ entailing not only the creation of additional files, but also some obscure behaviour already documented. [https://www.sqlite.org/wal.html <last visited: 21.02.2017>] Due to the latter and the fact that the desired files-exchange between platforms works best with monolithic databases, the default journal-mode (DELETE) is applied. Besides, according to the peculiarity of archaeological data remaining rather unchanged once created simultaneous reading and writing is barely expected.

3.4.4.4. ConnectionUtility

Though all components illustrated in Figure 7 are indispensable for a working application, the key building part definitely is the ConnectionUtility-class. Like the TreadManager-class (see chapter *2.4.4.1. ThreadManager, Assignment and LockFreeQueue*) this component is kept statically in the manner of the Singleton pattern [see Dooley 2011], for it needs to be accessible by all G$_{raphical}$U$_{ser}$I$_{nterface}$-components and must solely stand sentinel over database-connections. Consequently, all its public functions are static and constitute the only access-points to the actual private and static ConnectionUtility-object.

The significance of the component originates from its role as connector between data-model, databases, thread-management and G$_{raphical}$U$_{ser}$I$_{nterface}$ (see Figure 7). A typical task could, therefore, be: receiving commands from the GUI, acquiring necessary information from the data-model and/or glossaries, managing thread-safety while consulting databases and/or glossaries and finally executing operations immediately or launching assignments at the ThreadManager. Although commonly GUI-components consult the ConnnectionUtility as a means to have it accomplish tasks (concurrently), at some occasions the GUI directly launches concurrent assignments, so as to directly receive assignments' signals[65] and bypass the ConnectionUtility. Since the latter being called by those very assignments and functioning, thus, as

[64] see: https://www.sqlite.org/capi3ref.html <last visited: 21.02.2017>.

[65] ...which is possible due to GUI components inheriting from QObject and residing in the applications main-thread.

gatekeeper to critical resources anyway the ConnectionUtility is still able to guard the application against inconsistent state. Because the component holds a reference to the data-model-object, it is also responsible for delegating data-model related tasks (like changing the data-model currently in use), while guaranteeing the validity of the model. So irrespective of the various private members, functions and public constants, the ConnectionUtility's public functions can be grouped into the following topics discussed subsequently: SQLite/Spatialite, data-repositories, data-model, glossaries and indices. Two important internal concepts deserve mention, too, namely interaction with the master-database and thread-safety.

The Master-Database

Like mentioned in chapter *1.2.3. Conceptual Solution* and in regard of Figure 7 the application requires a place for permanent storage of critical information, since it needs to track registered data-models and -repositories. Because the application utilises sqlite-database-files anyway, one such is set up for this very purpose and consulted exclusively by the ConnectionUtility-component.

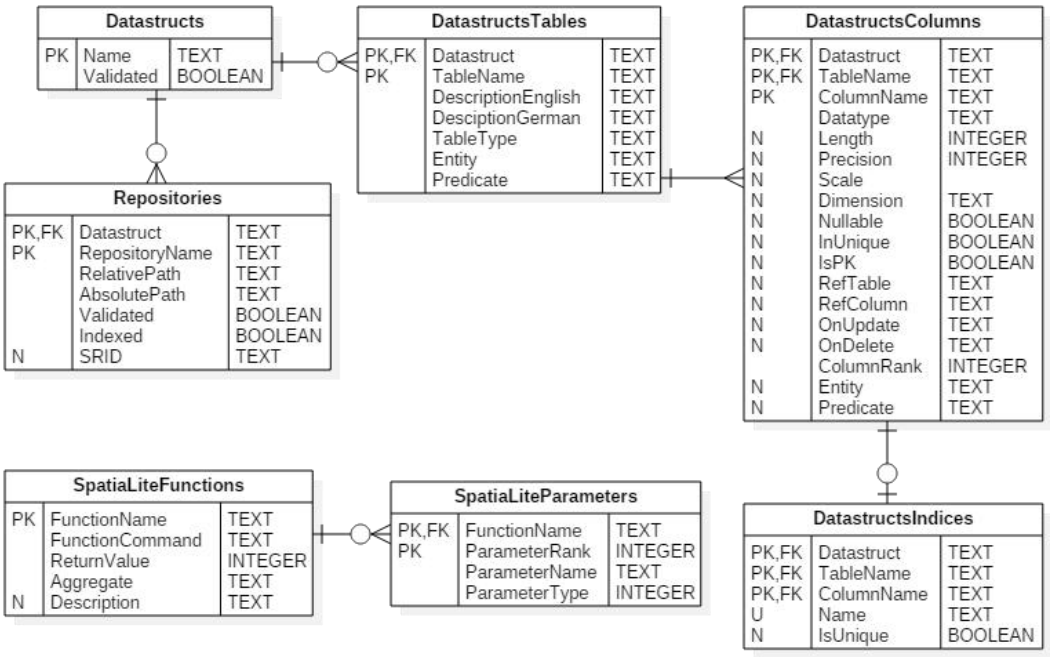


Figure 15

entity relationship model of MasterDatabase's structure

Figure 15 not only explicates the content of the master-database, but also unveils some concepts being subject to discourse in later chapters once more. To begin with, according to Figure 15 the master-database memorises entire data-structures as well as names and file-paths of repositories, thereby letting the application localize those resources, not to mention initialize the DataStructure-component without reading the data-model-document. Additionally, the database logs, if data-structures and repositories have already been validated for use within the current session. Besides the spatial-reference-system (SRID) being stored with the repository in the corresponding table, there is also a column called 'Indexed', the significance of which being discussed below. The tables SpatiaLiteFunctions and SpatiaLiteParameters are intended to hold information on SQLite/SpatiaLite-functions, in order to enable the application to offer them for use in database-transactions. As a matter of fact, the ConnectionUtility provides functions for storing and retrieving all the tables' data and as a means to speed up queries the master-database is fully equipped with database-indices.

Concurrency and Thread-Safety

Since due to the application's multithreading mission the master-database, repositories, glossaries and other resources demand coordinated thread-safe access - while not shifting that issue to the GUI -, a procedure was contrived allowing for several simultaneous readers, however only one exclusive writer to approach each of aforesaid resources. The undertaking represents, thence, an internal locking-mechanism to single files, while at the same time bearing a major challenge arising from the intention not to have a writer starve waiting for a constant rush of readers to end.

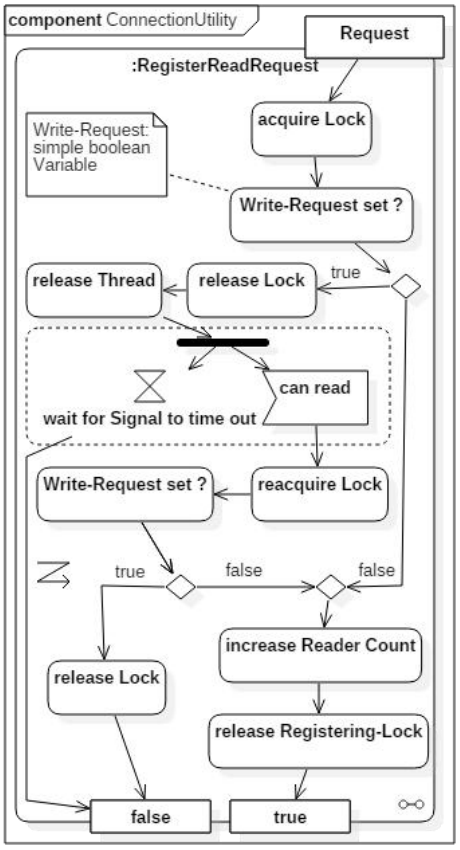


Figure 16

activity diagram of registering read-connection

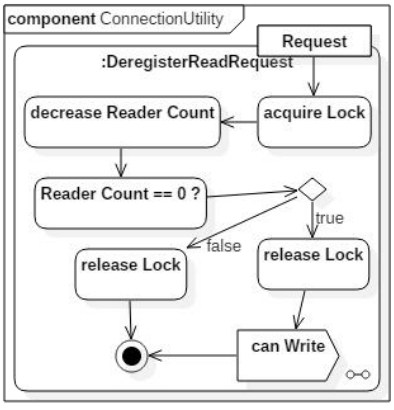


Figure 17

activity diagram of de-registering read-connection

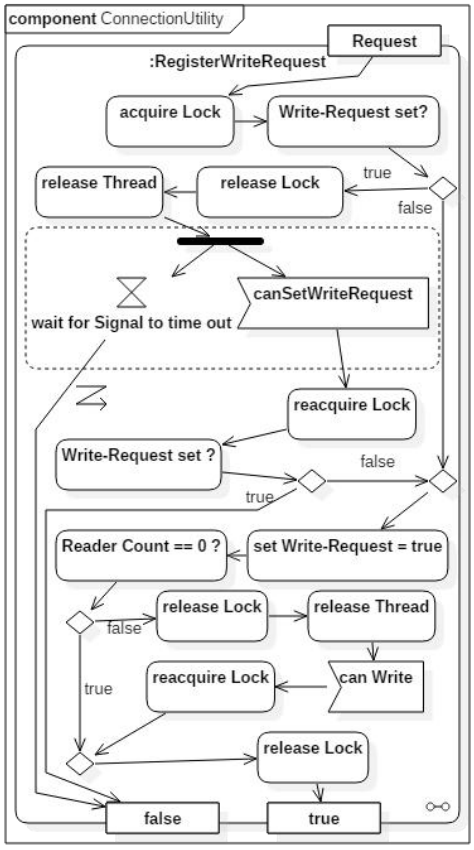


Figure 18

activity diagram of registering write-connection

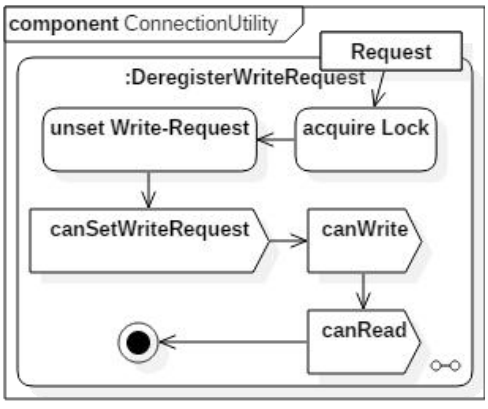


Figure 19

activity diagram of de-registering write-connection

The Activity-Diagrams of Figure 16 and Figure 18 illustrate the processes of registering the wish for accessing a certain resource, whereas Figure 17 and Figure 19 display the corresponding de-registering acts, more precisely the announcements of having finished resource-interaction. The boolean return-values, thereby, indicate the success of the attempt.

The mechanism comprising all four illustrated procedures involves one lock/mutex, three condition-variables represented in the figures by signals, a boolean Write-Request-variable and a Reader-Counter, in order to meet aforementioned expectations regarding a writer's precedence over readers. While there is actually no mutex or file-lock preventing resource access, as the applied locking only serializes the registering-process, the thread-safety is nonetheless preserved, as long as the actual file-interaction is enclosed by the corresponding registering/de-registering functions of Figures 16 to 19.

According to that principle, not only the master database is guarded against unwanted side-effects of concurrent access, but also glossaries and repositories, the latter two, however, in a rather dynamic fashion. This means that for the time of resource-interaction the above mentioned corresponding variables (mutex, condition-variables etc.) are stored in a map (once for each resource) and deleted, after interaction is finished.

In order to round up the picture on concurrency practised in the ConnectionUtility, it must be stated that further measures are taken and additional mutexes involved operating on a rather 'global' level assuring full thread-safety.

SQLite/SpatiaLite-Functions

SpatiaLite, even more than SQLite, comes with several build-in functions[66] applicable in database-transactions and essential for spatial tasks. Since the $G_{raphical}U_{ser}I_{nterface}$ provides a query-form (see chapter *2.4.5.6. QueryBuilder-Form*), those functions are kept at disposal in the master-database and are retrieved and stored via the ConnectionUtility.

[66] https://www.gaia-gis.it/gaia-sins/spatialite-sql-4.4.0.html <last visited: 30.04.2017>.

(Data-)Repositories

Because data-repositories are the major concern of the ConnectionUtlity, most functions center upon the following thematic categories, namely setting up the data-model in the database-file, validating the repositories' content, importing, registering, creating, exporting, copying, merging as well as removing repositories and finally actual database-transactions for data-management, not to mention several helper-utilities.

In doing so, those tasks act upon two different levels, namely files-level and database-internal, the most basic among the latter being the first two of aforesaid areas of application, for they both are employed - directly or indirectly - in several other functions. The setting-up of the data-model in a repository-file is, thereby, achieved by a single transaction containing all necessary Create-statements originating form the DataStructure-object, so that in a second step glossary-values kept in the XML-documents can be inserted. The validation of repositories, too, relies heavily on the data-model and takes advantage of SQLite's automatically generated metadata, which for this purpose is compared to data-model-data in the DataStructure-object. Optionally within this procedure, glossary-values are checked making the validation-process conceivably a relatively long-lasting task. Irrespective of conventional database-transactions - although complex since tailored for the GUI's needs - the ability of merging and copying database-files deserves mention, as they - together with repository-indices (see below) - break up the rather static separation of data into different database-files.

As indicated by the term of aforementioned first level of application, a lot of file-handling is involved due to the program's mission to herd a group of database-files. This includes not only keeping the directories of the application's external dependencies tidy and free of unwanted files, but also synchronising registered repositories with repository-files actually existing. Additionally, in the context of repositories most functions entail consulting the master-database, be it for querying the path to a repository, be it for changing the registration-status of a repository-file.

Since an essential part of herding a swarm of repositories is ensuring a correct and conformant data-model in all database-files, the means of choice is allowing repository-interaction only as soon as a repository has been validated, which impends each time the current data-model changes.

Data-Model

In regard of the data-model, respectively the DataStructure-object within the ConnectionUtility, it is not so much about the already mentioned functions informing about the details of the current model, but rather about the mechanisms for removing, importing, creating and changing data-models, the first two of which simply involving the existence of the corresponding $_{e}X_{tensible}M_{arkup}L_{anguage}$-documents in the designated folder as well as informing the application - respectively the master-database - about the changed facts (in order to remove entangled resources etc.). When creating a new data-model, however, the information on the new data-structure coming from a $G_{raphicl}U_{ser}I_{nterface}$-form is handed to the ConnectionUtility triggering a rather complex process:

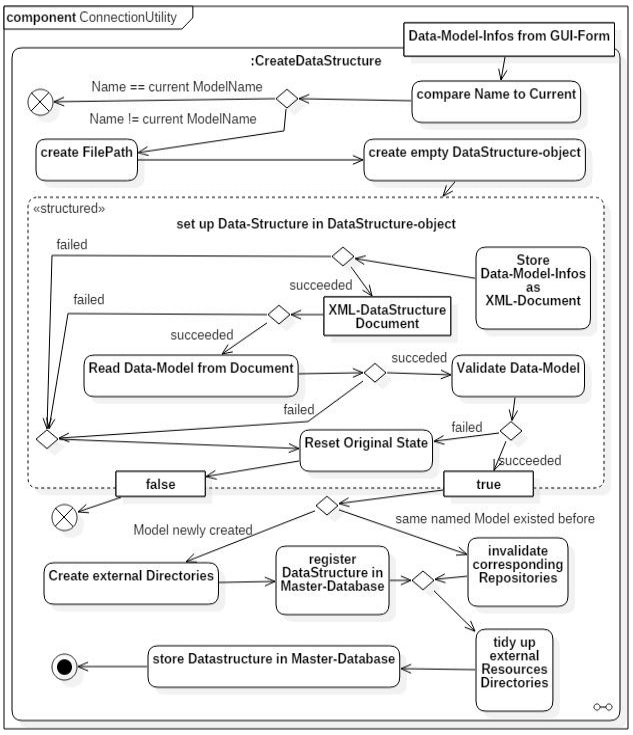


Figure 20

activity diagram of creating a new data-model

Although simplifying, Figure 20 roughly illustrates the process of creating a data-model preconceived in the GUI. It is worth noting that in executing the final step the data-model is stored in the master-database, in order to offer quicker model-access than reading the data-model's XML-document. Special focus also lies on the actions taken after the data-model was established at the DataStructure-object, for several further activities might be expected taking care of managing external resources. The latter, though, is not necessary due to the change of the current data-model including all essential checking:

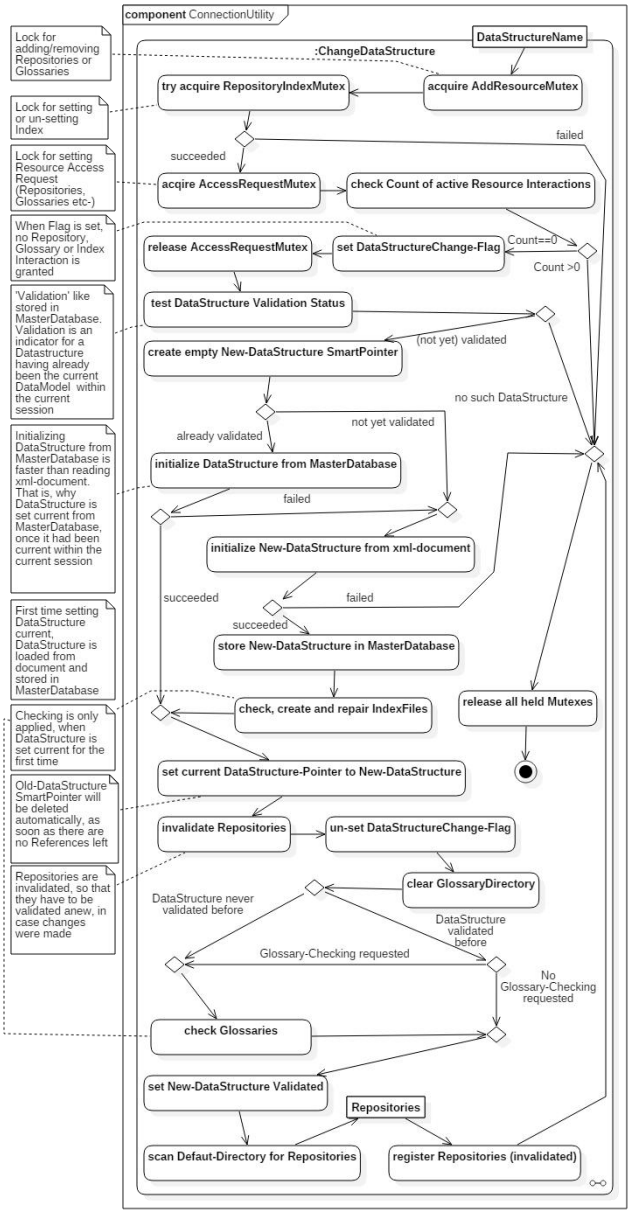


Figure 21

activity diagram of changing data-models

In Figure 21 not only the procedure of setting a data-model currently active becomes apparent, but also a special feature of the application, which is to say (explicitly) that only a single data-model can be active. Beside providing a glimpse of insight in the involved and rather complicated thread-safety-related locking mechanisms, the process displayed reveals, first of all, that the application takes advantage of tracking validation-events of repositories and data-models, in order to speed up data-structure-change as well as preserve repositories' data-model-integrity. One of the key actions, thereby, is storing the data-model in the master-database once set active (and validated) as well as retrieving it the next time it is requested, which is a lot faster than reading the data-model from the XML-document. Invalidating the repositories each time data-models change is necessary, in order to force repository-validation anew and therefore obviate external data-corruption.

Glossaries

Like outlined before, glossaries serve as vocabulary-providers to certain fields specified in the data-model. As long as the corresponding data-model exists, their values must be stored permanently in text-files complying with the rules of $_{e}X_{tensible}M_{arkup}L_{anguage}$, because for reasons of data-integrity those terms are transferred to corresponding tables in newly created repositories, so that all database-files share the same glossary-content.

The management of glossaries is straightforward, for ConnectionUtility's corresponding functions center upon reading and writing XML-glossary-files, keeping repositories' glossary-related content consistent, watching over the XML-files and even filling a glossary-file with values from a repository. Changes to the actual glossaries, however, can only be made, when there are no repositories registered, since that would immediately make repositories' content invalid.

(External) Indices

When talking about indices the term possesses a twofold meaning in the context of this application. First, the term relates to indices specified in the data-model and installed in the repositories, in order to speed up queries. Foreign-key-columns as well as spatial-columns are given indices automatically by the application without the need of coding them in the data-model.

The second kind of index is supposed to bridge the gap between the separate repositories to some degree, if the user regards it a desire. Therefore, as illustrated by Figure 2, Figure 6 and Figure 7, each data-model is granted database-files (henceforth called external indices or IndexDatabases) carrying the names of declared non-glossary tables. Each such IndexDatabases-file is able to contain tables named after corresponding repositories and filled with a copy of the values stored in the repository's table carrying the same name as the external index-database-file. Figure 22 is supposed to elucidate the principle:

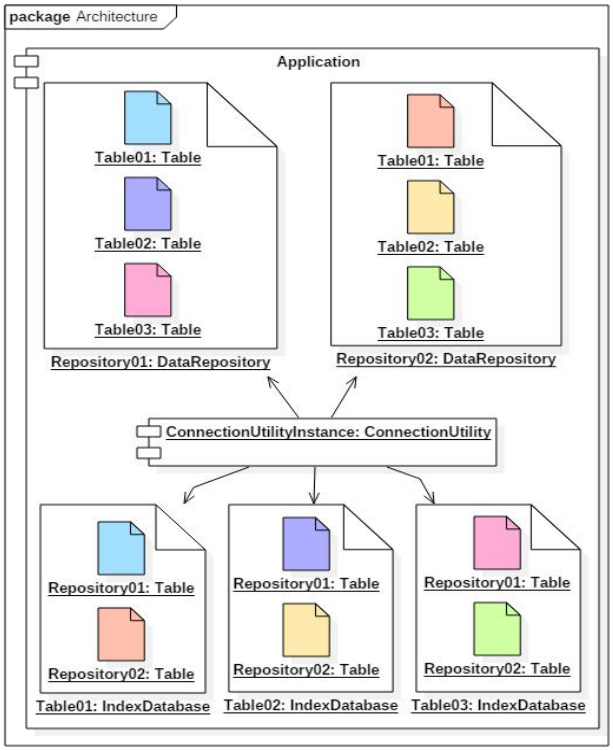


Figure 22

component-diagram with external indices

Therefore, adding a repository to that mechanism does not only almost double the data, for copies of the repository's non-glossary tables are allocated to the various IndexDatabase-files, but each such

external index combines the corresponding tables across all chosen repositories, in order to facilitate a kind of external index (as the name already implied). So, if repositories were regarded as vertical slices through the overall data, those indices could be considered the horizontal equivalent. The reason for keeping the data separated in different tables (named after their repositories) is, first of all, the convenience of having all data regarding a certain repository on hand immediately. This allows not only for creating, deleting and counting entries as well as tables (respectively indexed repositories), but should also speed up queries due to the option of specifying the affected tables and consequently scale well with the number of indexed repositories. Although highly dynamic databases may suffer from performance penalty as a consequence of synchronising with such index, the archaeological data is not expected to be that dynamic - like already stated. Besides, repositories can be registered as well as de-registered for the external indexing at will. As a means to further enhance performance of external indices, though, the classical database-internal indexing is also employed[67] heavily, whereas BLOB fields are not supported in the external indices, so as to keep database-file-sizes small.

The practical goal pursued with the external indices is, on the one hand, fabricating an instrument for quick keyword-like-searches and data-exploration across the indexed repositories, on the other hand having thematically related geometries stored in the same table for spatial analysis.

A capacity of SQLite not yet introduced, however highly involved in transferring data between databases - like in the external-index-topic - is the ability of attaching[68] databases to existing database-connections, so as to have those files act like a single database (requiring minor adaptations in SQL-statements). The application, not to say the $G_{raphic}U_{ser}I_{nterface}$ in cooperation with the ConnectionUtility, provides this feature also for affiliating two different repositories as another means to overcome overall data-segregation (see chapter *2.4.5.6. QueryBuilder-Form*). The major downside of the attach-method, though, is the limited number of attachable databases[69] having led not only to the external index mechanism, but also to the applications limit of linking only two repositories at a time (see chapter *2.4.5.6. QueryBuilder-Form*).

For the practical implications of this chapter's fully implemented functionality on the applications capabilities and behaviour see the following chapters discussing the $G_{raphical}U_{ser}I_{nterface}$ from a users perspective.

2.4.5. Functionality

After having led through the desktop-application's internal structure populating the functional layers underneath the $G_{raphic}U_{ser}I_{nterface}$, this treatise continues outlining the GUI's behaviour in respect to the applications contribution to the overall objective. Since the involved functionality, thereby, mostly centers upon Qt's widgets and signal-slot mechanism discussed on the corresponding website[70], this chapter - instead of highlighting the widgets' rather straightforward functional interplay - focuses on the implications of bringing forth the program's capabilities to the benefit of the archaeological data-handling-process, which is then intended to unleash the (spatial-) data's full potential.

As a single general side-note on the widgets interaction it is to be stated that a recurring principle of the GUI is the application of a kind of model-view-controller pattern[71] employed, wherever lists or tables of stored data are to be displayed, for Qt offers suitable classes [http::/doc.qt.io/qt-4.8/model-view-programming.html <last viewed 28.02.2017>] applicable those very situations.

[67] Tests with a single fully indexed table filled with about 50 million entries give rise to the hope, that at least up to that count the performance of a SQLite-database is nothing to worry about, as long as the query allows the internal indices to exert their power.

[68] see: https://www.sqlite.org/lang_attach.html <last visited 27.02.2017>.

[69] see: https://www.sqlite.org/limits.html <last visited 28.02.2017>.

[70] http://doc.qt.io/qt-5/ <last visited: 04.03.2017>.

[71] see for example Dooley 2011.

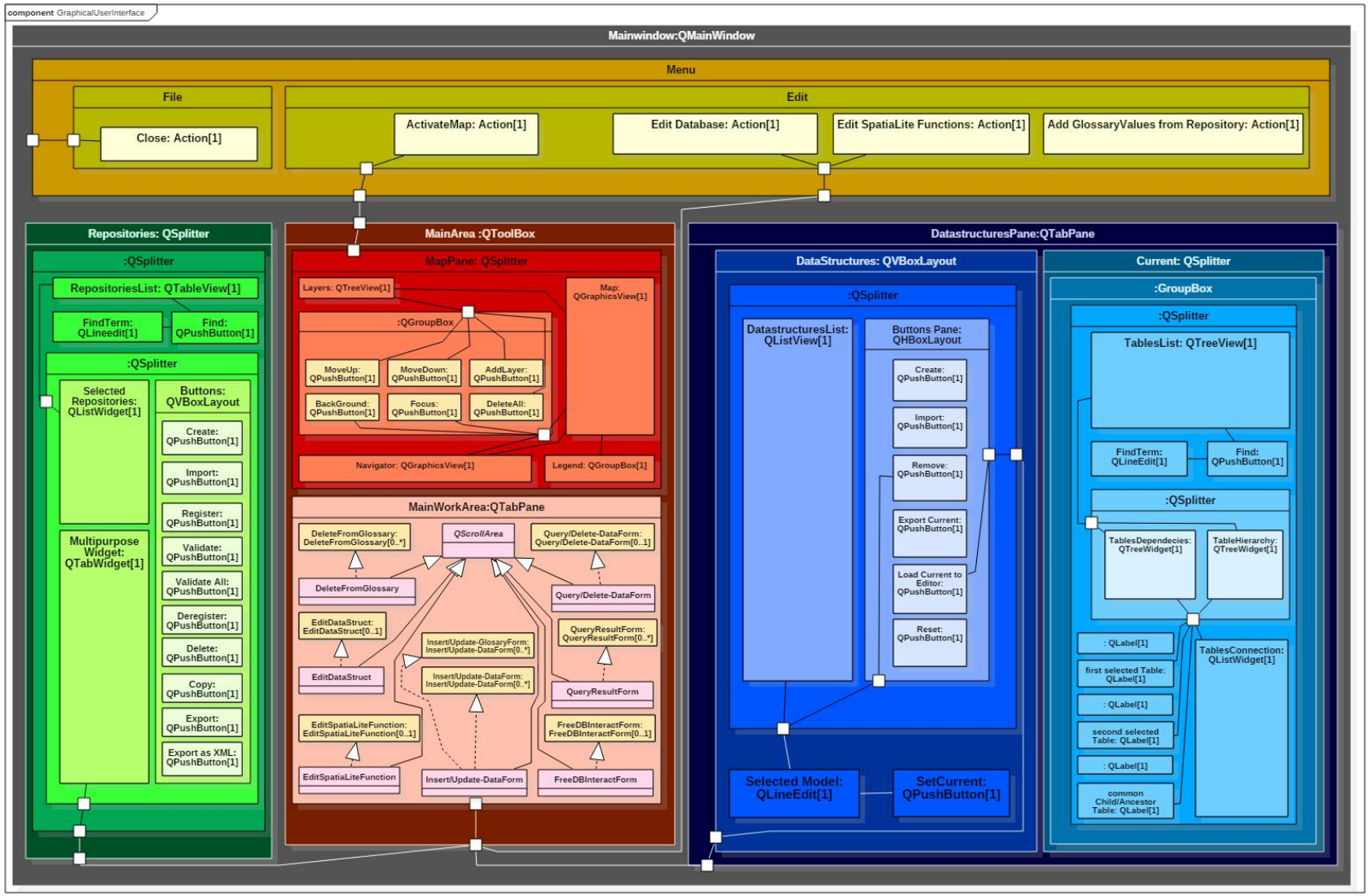


Figure 23

composite structure diagram of the GraphicalUserInterface

Since Qt-Designer is involved, the Widgets of the program are not actually nested in the shown way from the programmer's point of view. The figure is nevertheless an approximation of how the GraphicalUserInterface would be structured applying classical coding exclusively and of where inter-widget communication has to happen. Additionally the employed Qt-Classes become apparent, whilst the location of the parts resemble the actual positions of widgets

Like illustrated by Figure 4, Figure 5, Figure 23 as well as chapter *2.4.2. GUI-Overview* the GUI is grouped in certain assemblies of widgets collectively affecting a common functional aspect, like, for example, the data-model. Several of those widget-groupings, namely those permanently active, were already identified in Figure 5, whereas others, like the form for data-input, are to be activated dynamically. This chapter intends to walk through those groups one at a time, in order to highlight the role of each such functional group in the context of this paper's goals of handling spatial archaeological data.

2.4.5.1. Managing Data-Models

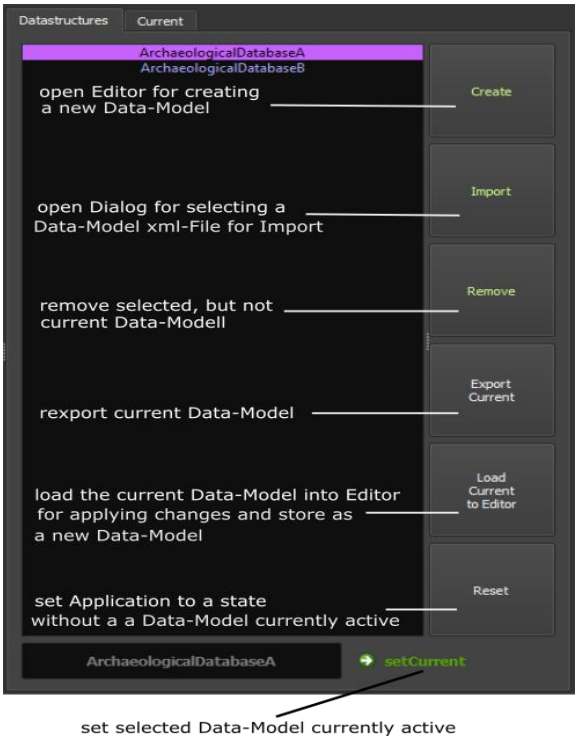


Figure 24

data-structures pane from GUI's right-hand area

A data-model like presented in chapter *2.2. Data-Model* is highly expected to be subject to modification as time progresses, as much as the archaeological user is likely to wish for handling diverse data-models at the same time. Since, as a matter of fact, the model, therefore, must not be static or statically connected with the application out of reasons of practical applicability, the GUI offers all necessary functions for modifying or switching between data-models in a single pane. In doing so, the rest of the program only takes into account the capabilities and resources (e.g. glossaries) associated with the current model. Since Figure 24 is, for its most part, self-explanatory, it suffice to mention two aspects, first of which is the setCurrent-button triggering the process highlighted by Figure 21. The second aspect regards the Data-Model-Editor-Form activated by two of above shown buttons. Just like the current data-model is not displayed in the data-models-list, so as to prevent accidental deletion or modification, the editor assists the user - respectively the archaeologist - in designing models safely and hence does not store new data-models with the same name as the current, so that the editor only either creates a new model or replaces a model currently not active.

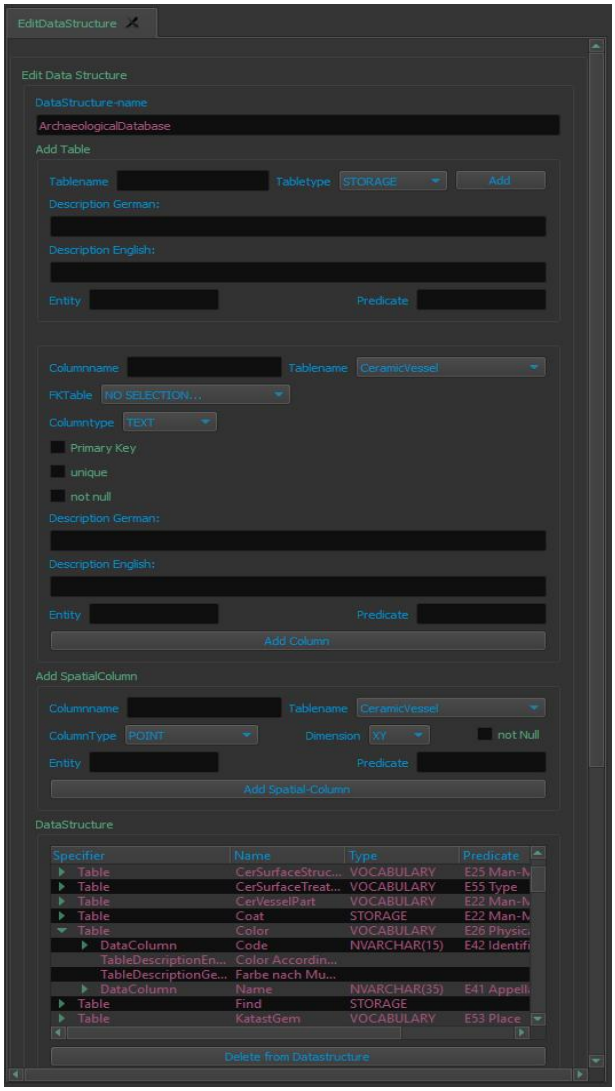


Figure 25

data-models-editor on multi-purpose work surface

The data-model-editor enables the user to generate or modify data-models from within the application, so that the data-model-scheme does not even have to be consulted attempting to investigate the correct syntax for a data-model file. As this only accounts for an archaeologist not being fluent in XML, the most important characteristic of the editor, however, is its role as first instance of validation providing a highly dynamic form allowing only valid table-, index-, column- and foreignkey-definitions. This means, for example, that an index can only be defined for a column actually existing, that, when removed, also deletes the index from the model. If a data-model is intended to be based on another, the already validated current model can be loaded into the editor, in order to apply modifications and store the results in a new file.

A roughly outlined work-flow of setting up a new model on the editor, hence, would comprise defining tables, adding them to the tables' list, adding data- or spatial-columns to the tables, defining indices to apply in each data-repository and finally triggering the multi-threaded process of trying to establish the new data-model. The latter is achieved by handing the data over to the internal components via assignment, like outlined in the chapter *2.4.4. Main Internal Structure*, while the results are transferred back to the GUI (-thread) via signal-slot mechanism telling the processing window meanwhile sprung-up to either display occurred errors or disappear.

Once created, the model can be exported and shared with fellow archaeologists, whereas, as far as receiving data-models is concerned, the import-function is in charge requiring only a file-path to a valid data-model-XML-file, while consequently enabling a multitude of people to easily apply the same data-structure simply by text-document-exchange thence fulfilling a major requirement.

Because consistency with external resources is taken care of when changing data-structures and validating repositories, the editor simply needs to trigger the creation of the data-model's XML-file.

2.4.5.2. The current Data-Model

Since information on the current data-model is essential, the application dedicates a complete pane to that subject on the GUI's right hand side:

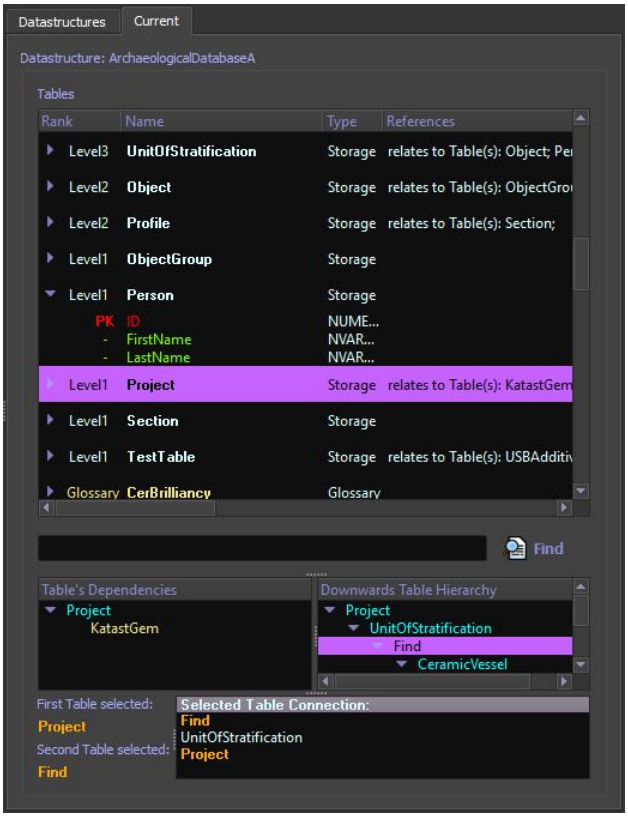


Figure 26

current-datastructure-pane

As apparent from Figure 26 the information on the current data-model comprises, first of all, a TreeView of tables and columns together with all parameters specified in the data-model document. The tables are ordered by their type - storage or glossary - and by their foreignkey-columns, which means, that a table of a certain level references only tables on levels numbered less. This way a sequence builds up informing the user to first add entries to tables more basic on lower numbered levels, before filling higher levelled tables depending on the former with their foreignkey-columns. The same concept is additionally expressed by the two TreeWidgets named "Table's Dependencies" and "Downwards Table Hierarchy" located below the first TreeView. The latter two change their content dynamically according to the selected table and are themselves interactive, which means that the user simply clicks two table-names, in order to have the program exhibit the tables' connection consequently displayed at the bottom-area of the pane. This

happens not just for demonstration-purposes, but is the base for selecting tables to be queried, so that the program can take care of constructing the correct table-joins in an SQL-statement. Like mentioned above the applied technical processes mostly involve the widgets' common functionality, except for the procedure of retrieving the data-model for display in the TreeView in such a multi-threaded way that an assignment (see *2.4.4.1. ThreadManager, Assignment and LockFreeQueue*) takes care of filling a QStandardItemModel in a worker-thread, in order to assign the model to the TreeView later employing the GraphicalUserInterface's thread. As it turned out, only the thread running the GUI can directly interact with the widgets, therefore, requiring certain rather complicated algorithms.

2.4.5.3. Handling Repositories

At the left-hand side of the GraphicalUserInterfaces the pane for handling the various data-repository-files is located fulfilling a basic requirement of this endeavour:

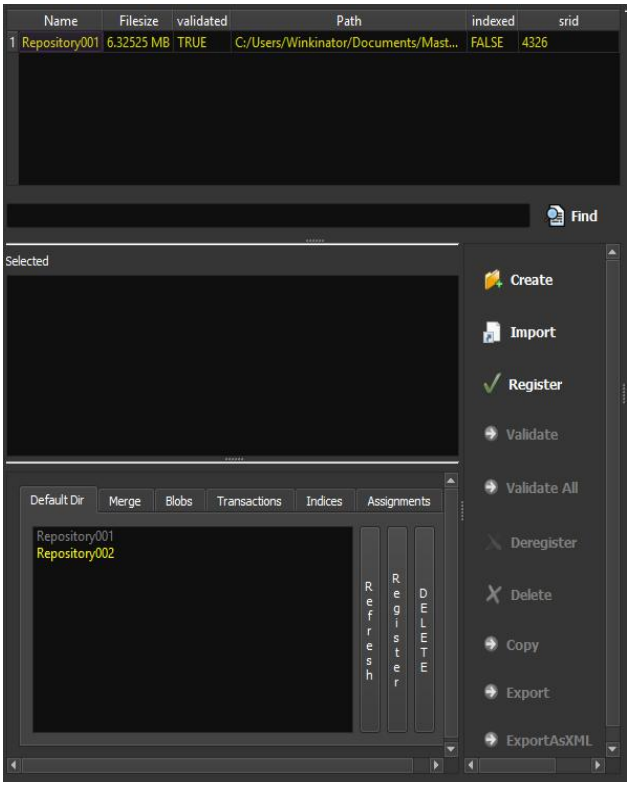


Figure 27

data-repositories pane

Certainly a user must keep track of the data-repositories associated with each data-model, so that the topmost widget of the pane illustrated in Figure 27 representing a QTableView-widget displays all registered repositories. It is to be noted that the table features the columns "indexed" and "validated", the latter of which informing about the capability to interact with a certain repository-file's content. Each repository has to be validated via clicking one of the Validate-buttons, before it can actually be selected (visible in the Selected-QListWidget) or even remotely interacted with. This mechanism guarantees valid data-models in the repositories, while no automatic long-lasting overall validation-process involving all repositories makes the user wait unnecessarily.

At the bottom of the pane several functional groups are bound together in a QTabWidget, one of which is an overview of the repositories' default directory being, thus, at the user's disposal in terms of re-registering or deleting repositories currently not registered. Right to that there are all buttons concerning the handling of repository-files.

Creating, Importing or Registering a Repository

The topmost button shown in Figure 27 triggers the algorithm for creating a new empty data-repository illustrated by Figure 28. The image additionally displays all the involved dialogues, one of which being a processing window with its twofold purpose of bridging the time of processing before outputting the results. Hence, it exemplifies the interplay of the implemented multi-threading-mechanism with Qt's signal-slot paradigm. Another dialogue in Figure 28 allows for selecting a spatial reference system to apply in all spatial columns of a repository thus exploiting the fact that in the scope of an archaeological project all spatial data most certainly adheres to the same reference system anyway. The values of the dialogue, thereby, originate from one of the internal metadata-tables of the file sample.sqlite, which, as Figure 28 elucidates, mainly serves as source for new repository-files, for initializing spatial meta-data in a newly created SQLite-database would take up to several minutes.

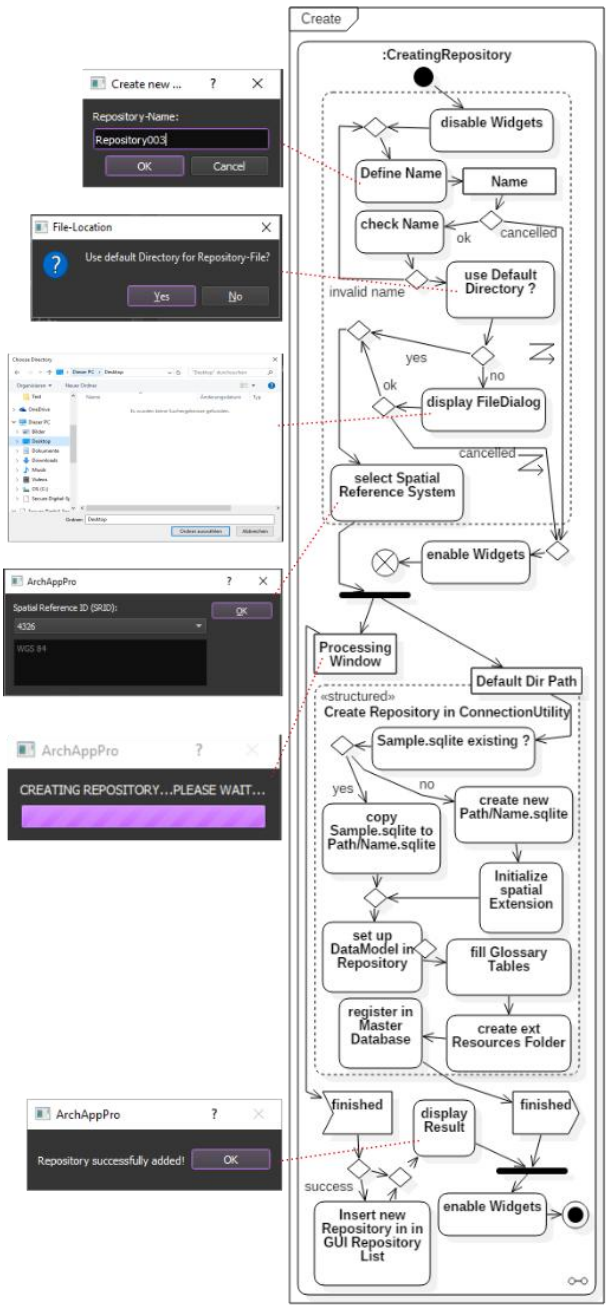


Figure 29

activity diagram of creating a repository

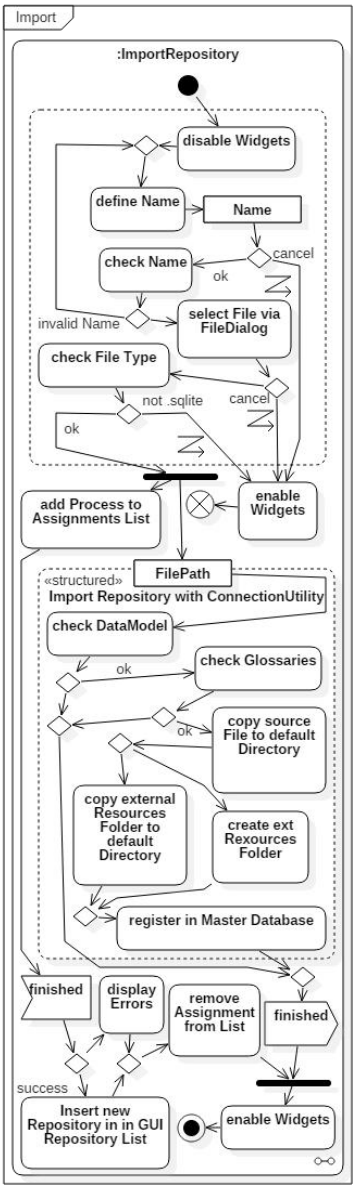


Figure 28

simplified activity diagram of creating a repository

One aspect not yet discussed, however indicated by Figure 28 is the fact that each repository in the default directory is accompanied by a folder capable of storing external files referenced by the repository. Such files could be 3D-Models or photographs and are accounted for in the major repository-activities, so that the creation of the repository may entail creating the corresponding folder as well as exporting a repository copies the database-file together with its external files.

Figure 29 displays the simplified activity-diagram of importing a repository to the current data-structure. Beside certain similarities with the process of creating a repository it is to be noted that, if the concerned database-file is accompanied by a folder with the same name, the folder is considered the external references container of the database and is, thus copied to the new destination. Since checking glossaries is presumably a long task, another mechanism not yet introduced in the course of this paper is applied, namely running long term tasks (see chapter *2.4.5.10. Long Term Assignments*).

Because registering a repository is a process rather similar to importing, no activity diagram is provided here. It must be noted, though, that both methods are essential as a means to meet the requirements in terms of data-exchange.

Validating and Removing Repositories

About the buttons and processes of validating and removing repositories illustrated in Figure 27 there is only little to say. Removing, be it de-registering or actual removal, involves deleting the corresponding entry in the master-database,

updating the repository-list, deleting corresponding tables in external indices and finally possibly deleting the actual file-resources, while a window with processing bar keeps the user from interfering with the process currently running.

Validating repositories is from the user's point of view likewise straightforward, as the program triggers the concurrent checking of the repository's structure and - optionally - the glossaries' values. Again, a dialogue with processing bar bridges the time of waiting until the processes' results are displayed in the very same dialogue. If a repository is found invalid, it is not removed, but can only be interacted with in terms of removal, however, not for transactions, so that the data is not lost. As outlined before, the validation-mechanism is a core principle for preserving a consistent structural data-state, while no time is wasted unnecessarily in unwanted checking of all data-stores, so that the application scales well with the user's needs. Repository validations always stand at the beginning of all database-interaction.

Copying and Exporting

Copying and exporting are rather similar tasks, except that copying puts the result into the repositories' default directory, while exporting lets the user select the destination. Both, though, copy the database-file together with its external resources folder and store them under a specified name. Since splitting a repository in two separate files is a complicated procedure due to the issue of where exactly to separate the data, copying together with subsequent deletion of unwanted data is an adequate workaround thus leading to the copy-function. Exporting, on the other hand, aims at contributing to archiving and exchanging data.

Exporting a repository as $_{e}X_{tensible}M_{arkup}L_{anguage}$-file, as the lowermost button of Figure 27 proclaims, is in some way similar to the exporting just outlined, but differs in two major points being alphanumeric values - among them the geometries - getting assembled in XML-documents and BLOB entries being stored as separate files. Due to the possibly large number of entries the process not only is likely to be a long lasting activity, but the XML-files might also grow large in size, which is why the content of each table is spread across several files numbered in ascending order. Here is the corresponding schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.thomaswinklehner.at/
ExportedData"
xmlns="http://www.thomaswinklehner.at/ExportedDa
ta"
xmlns:tw="http://www.thomaswinklehner.at/Exporte
dData" elementFormDefault="qualified">
 <xs:element name="Table">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="Row" minOccurs="0"
        maxOccurs="unbounded">
     <xs:complexType>
      <xs:sequence>
       <xs:element name="PrimaryKeyColumn"
       minOccurs="1" maxOccurs="unbounded">
        <xs:complexType>
         <xs:simpleContent>
          <xs:extension base="xs:string">
           <xs:attribute name="Name"
            use="required" type="xs:string" />
          </xs:extension>
         </xs:simpleContent>
        </xs:complexType>
       </xs:element>
       <xs:element name="Column" minOccurs="0"
        maxOccurs="unbounded">
        <xs:complexType>
         <xs:simpleContent>
          <xs:extension base="xs:string">
           <xs:attribute name="Name"
            use="required" type="xs:string" />
          </xs:extension>
         </xs:simpleContent>
        </xs:complexType>
       </xs:element>
      </xs:sequence>
     </xs:complexType>

     <xs:unique name="ColumnUniqueness">
      <xs:selector xpath="tw:Column"/>
      <xs:field xpath="@Name"/>
     </xs:unique>

    </xs:element>
   </xs:sequence>
  <xs:attribute name="Name" use="required"
type="xs:string"/>
  </xs:complexType>

  <xs:unique name="PrimaryKey">
   <xs:selector
xpath="tw:Row/tw:PrimaryKeyColumn"/>
   <xs:field xpath="."/>
  </xs:unique>

 </xs:element>
</xs:schema>
```

So the result of this procedure is a folder of a specified name containing directories named “Tables” and “Files”, first of which comprising directories named after the repository’s tables (glossaries excluded). Within them aforesaid XML-files describing all the alphanumeric values of the particular table together with according BLOB-values are stored, while within the Files-directory all the repository’s external resources can be found. Whereas the common kind of data finds its way into the documents unmodified, spatial data is converted to GeographyMarkupLanguage, in order to have it intermingled with the textual data.

Consequently in combination with the data-model and the glossaries this exporting-mechanism is to meet the objective of long-term data-storage facilitating not only readability, but also accessibility by any technology capable of reading structured text-documents.

2.4.5.4. Inserting/Updating Data

In terms of inputting data - comprising also updating column values - the application intends to provide the most basic mechanism, in order to leave specified import methods like reading data from surveying-devices to future endeavours. Therefore, inserting as well as updating data into a repository both employ a form (opened on the multi-purpose-work-area via selecting a repository as well as a table) supporting the user in his ambition to input only structurally valid data, so as to generate a corresponding SQL-statement.

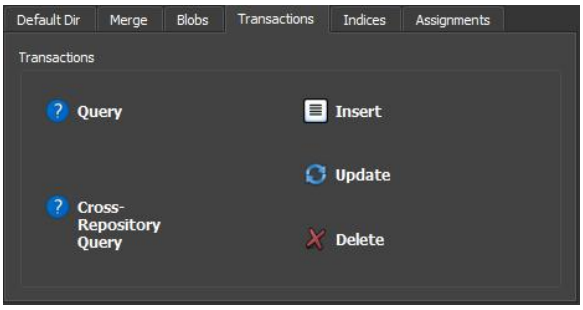


Figure 30

transactions tab of multipurpose-widget

This means that for all columns of the table data-fields are created taking into account the corresponding data-types (see for example Figure 31) as well as entailing several special features discussed subsequently. In order to update a table-entry, a similar form is provided additionally prompting the user for a condition.

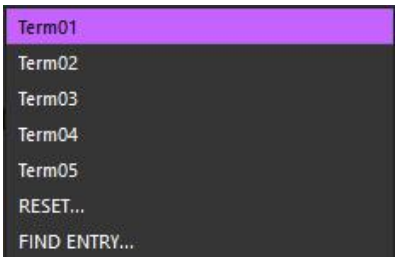


Figure 31

dropdown-menu

ForeignKey-columns are cast into the form of self-engineered dropdown-menus inheriting from QComboBox (Figure 31), but since a large number of entries in the referenced table may cause the program to crash, the count of values shown is limited leading also to a quicker responding widget. As a means to view all values nevertheless the items “PREVIOUS”, “NEXT”, “RESET” and “FIND ENTRY...” located at the bottom of the values-list coordinate browsing through all values, while the latter two even facilitate a keyword-search. While acquiring the values, the widget does not detain the program, since the process is executed in a different thread and the widget simply disables itself until it is ready again.

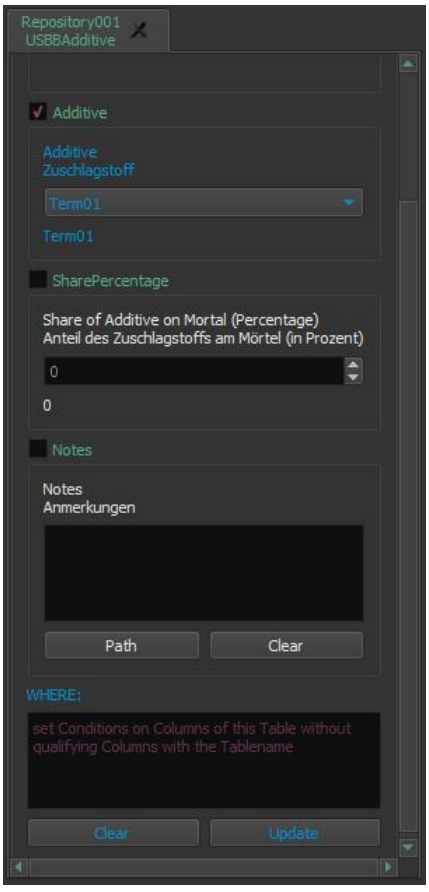


Figure 32

update-form

Like visible at Figure 32 normal text-fields provide a button named “Path”, that opens a file-dialogue for selecting a path, so that consequently the program can copy that file to the repository’s external files directory after having completed the

database-transaction. The user is prompted for a file-name, which is the value being stored in the text-field. This way, external files are linked to the alphanumeric data, as long as the database is stored in the applications default-directory for repositories.

Since the archaeologist is not assumed to know about textual representation of geometry, a major concern is having the spatial data stored in the database.

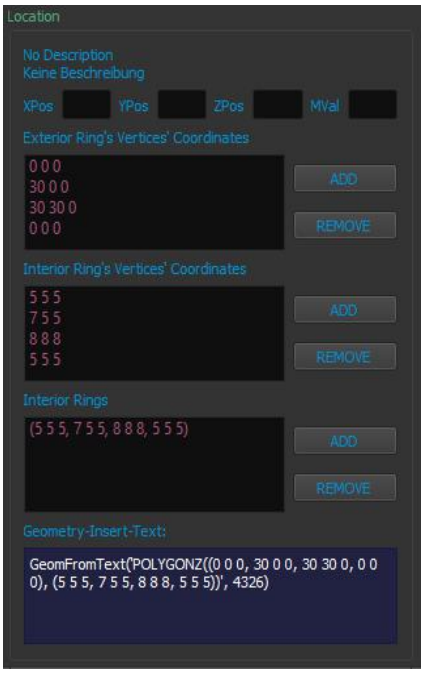


Figure 33

geometry-field

Although very basic the widgets of Figure 33 exemplifying the input-mask for polygons achieve to form a valid term ready for being employed in a SQL-database-transaction. The form allows for the remaining geometry-types to be defined in a similar manner each guaranteeing a valid resulting $W_{ell}K_{now}T_{ext}$-term. Although storing a file as BLOB in a SQLite-database is a rather challenging task for the application, for it has to be serialised and loaded into $R_{andom}A_{ccess}M_{emory}$, the user simply has to select a file via file-dialogue. For all other fields in the form it suffice to provide a suitable standard-widget and an event handler, that checks for valid input. In order not to have forms interfere with each other, the application only allows to open one form per table.

2.4.5.5. Inserting/Updating Glossaries

When there are no repositories currently registered, the user can modify glossary-values in a form, since it is up to the validation of repositories to unveil inconsistencies with glossaries. The form is somewhat similar to the one just described, whereas due to the limited data-type-range BLOBs and geometry are of no concern here. Also, the form does not generate an SQL-statement, but collects the entries in a treewidget, in order to write them to the glossary-file.

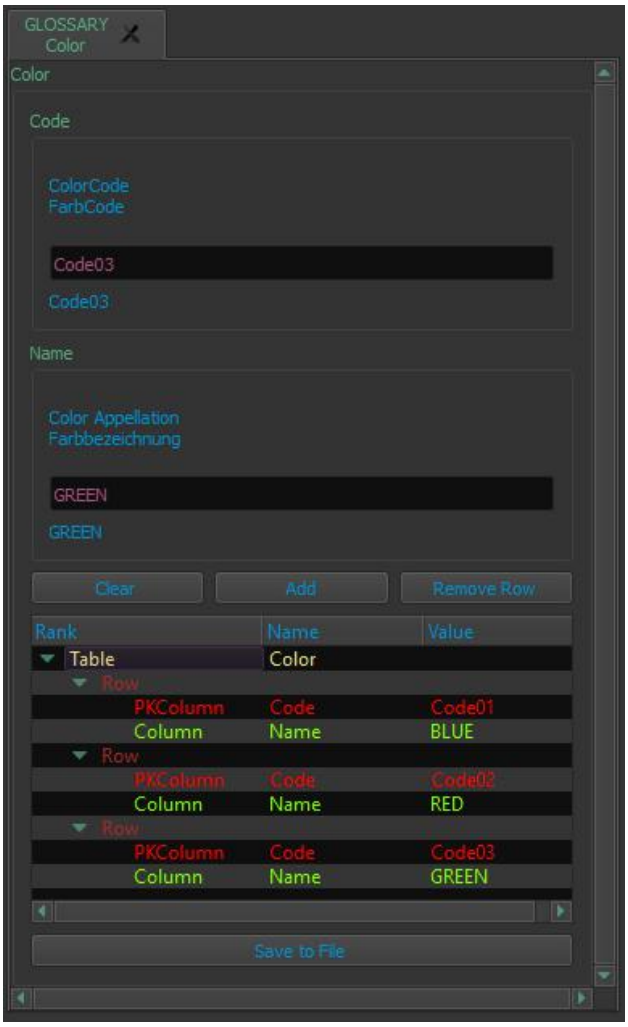


Figure 34

form for inserting glossary-values

Like all major operations storing the glossary-data is done applying multithreading, when handing the values over to the ConnectionUtility-component, which outputs the data employing the technology already discussed in chapter *3.4.4.4. ConnectionUtility*.

2.4.5.6. QueryBuilder-Form

As one of the leading overall objectives is retrieving data from repositories, special interest rests with the applications capability of querying the various database-files - in this case - facilitated by a form called QueryBuilder.

Since the archaeological user is not assumed to be an expert on SQL, implementing a QueryBuilder is preferred over having the user input the queries textually. In order to start the form, one or more repositories must be selected together with a table - respectively a tables' connection (see *2.4.5.2. The current Data-Model*), so that the form is able to join the tables correctly. The composition of the form itself is inspired by the structure of a SQL-SELECT-query. When the query is finally ready, it is sent to all selected repositories concurrently, so that multiple databases are queried at once.

What follows are illustrations of the QueryBuilder-form and corresponding explanations:

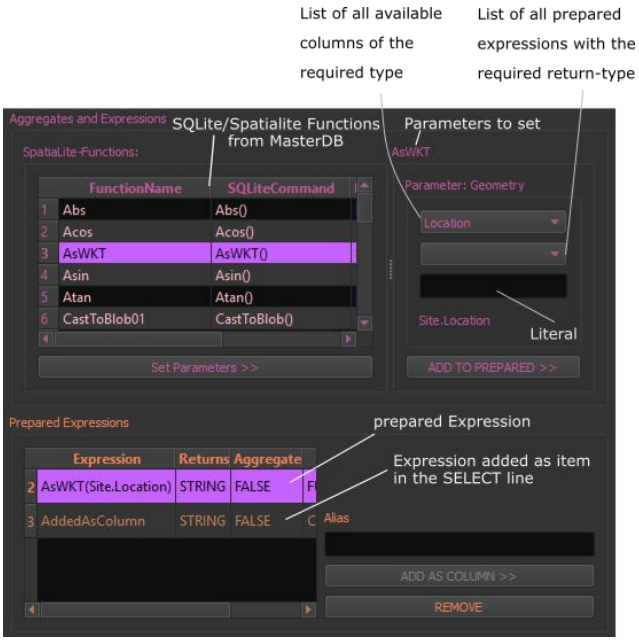


Figure 35

part 1 of the query-form: functions of the database

At the top of the form (see Figure 35) the user can assemble expressions made up by built-in functions of the database-system, columns of the selected table(s) and other expressions. The function's declarations, thereby, originate from the master-database. Due to careful design as well as the form making use of Qt's Model-View-classes only valid expressions can be constructed as a means to be employed in the SELECT- or WHERE-area of the query.

The subsequent Figure displays the widgets responsible for assembling the columns sought to query as well as the items effecting ordering and grouping.

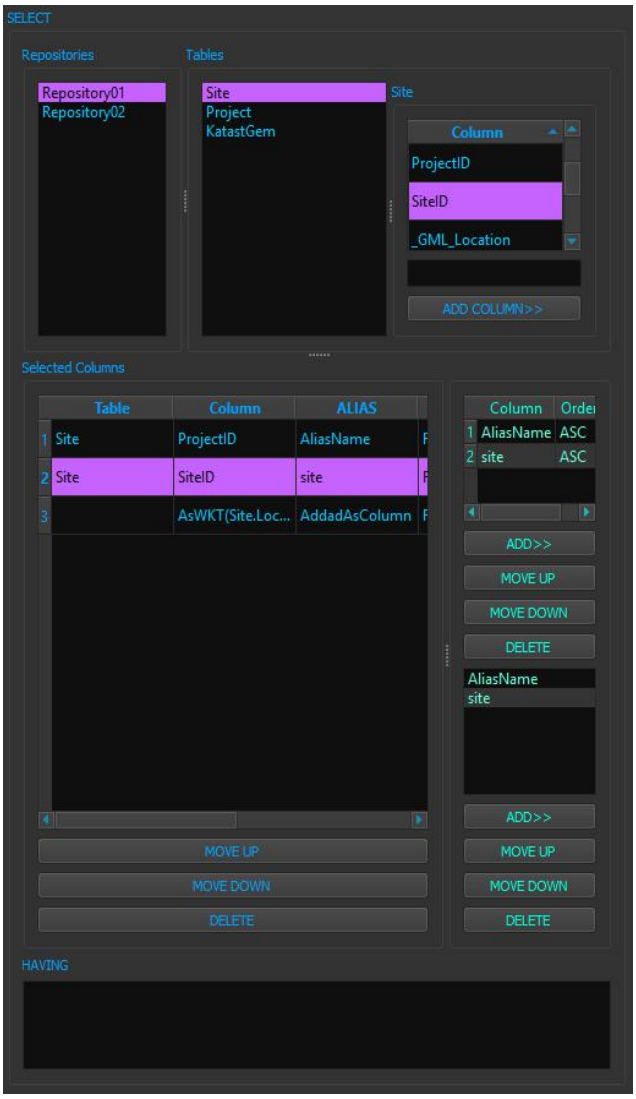


Figure 36

part 2 of the query-form: the select-area

The central component, however, is the list of selected tables (see Figure 37), that may be extended on the user's behalf. The list of corresponding columns is updated with each table added. The tables chosen initially by having generated a tables' connection (see *2.4.5.2. The current Data-Model*) are automatically joined in the resulting SQL-statement, whereas all additional tables must provide their own JOIN-terms.

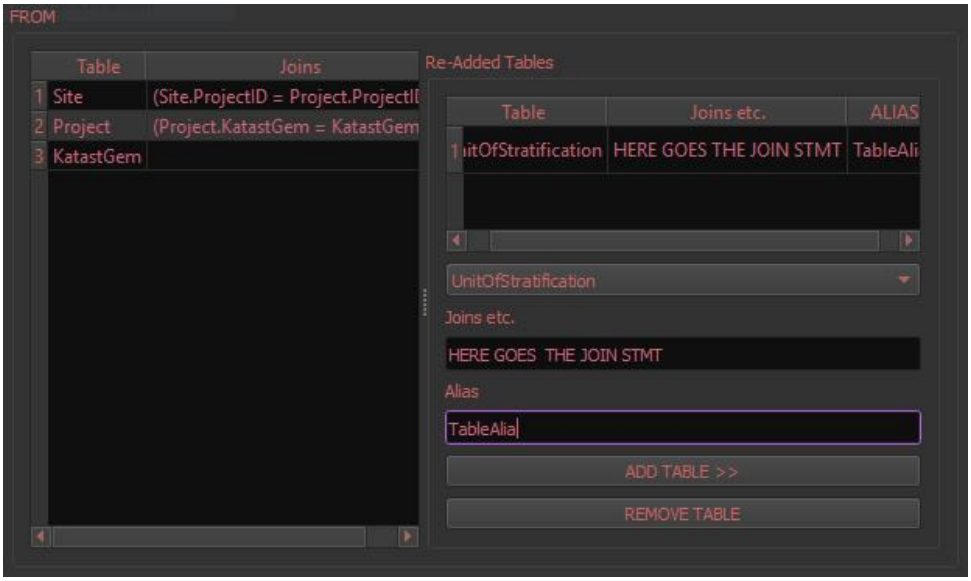


Figure 37

part 3 of the query-form: the from-area

The capability to add other tables than those initially selected is very convenient, when it comes to querying the spatial-index-tables, that possess the power to speed up spatial queries dramatically.

Below the tables-area a pane is situated responsible for building conditions later cast into the WHERE-term of the SQL-statement. Because it does not bring anything new to the discussion in conceptual terms, a corresponding figure is omitted here. The pane, however, has the ability to open the QueryBuilder-form in a modal dialogue, so that a subquery can be defined, which is then built into the SQL-statement of the main form.

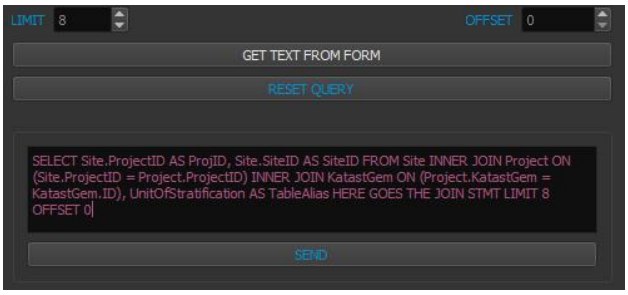


Figure 38

part 5 of the query-form (part 4 is not displayed)

At last, the user can have the form create the SQL-statement, which in any case is furnished with a LIMIT and OFFSET clause, in order not to overstrain the system (see Figure 38). The SEND-button , of course, triggers the query-process.

If the user selects exactly two repositories, the special method hinted above (see 'Indices' at chapter *3.4.4.4. ConnectionUtility*) for querying across two separate database-files becomes available by clicking the button named Cross-Repository-Query (see Figure 30). Thereupon the QueryBuilder opens like just illustrated above, but the names of the repositories, first of which now is called "main", are prepended to tables and colums, so that they can be identified unambiguously. Since this multiplies the number of entries in the columns' list, the reason for the limitation of two repositories becomes apparent. The rest of the procedure, though, follows the pattern described above.

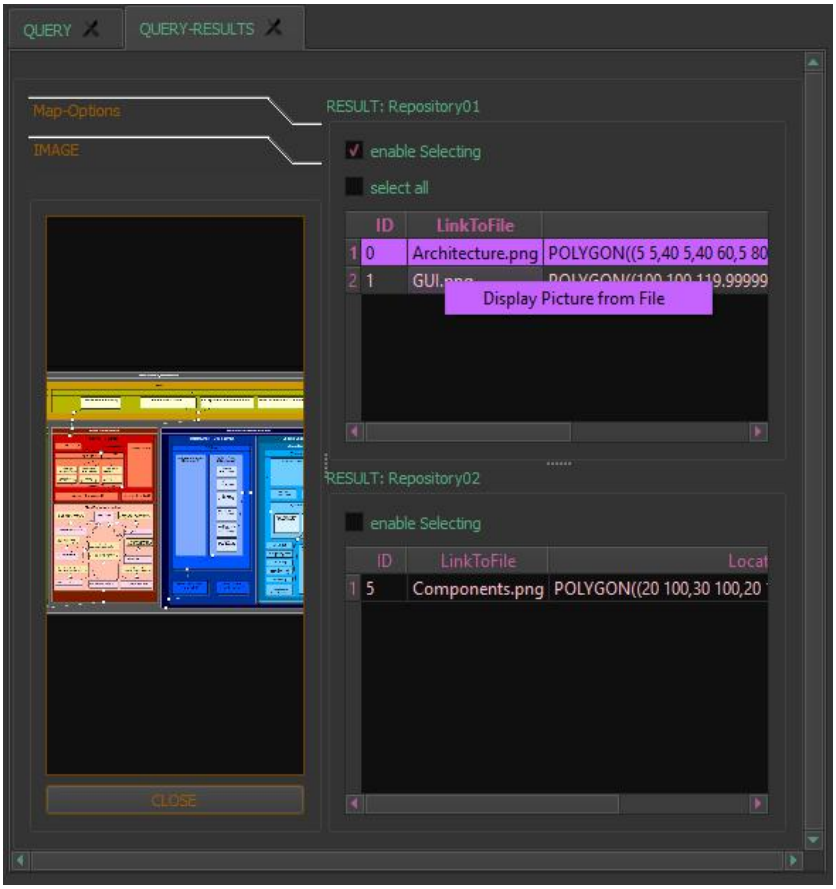
This method represents one of the essential features intended to bridge the separation of database-files.

2.4.5.7. Query Results

When the query is finally triggered, assignments are packaged for concurrent execution (for assignments see chapter *2.4.4.1. ThreadManager, Assignment and LockFreeQueue*) each conducting the query on a different database. Conveniently, the application does not block meanwhile, because only the tables displaying the results are disabled, as long as the worker-threads acquire the data to fill in, while the GraphicalUserInterface's thread is free to build up the results-form and do its day-to-day business. No window displaying progress keeps the user from interacting with the application, although the results-tab can not be removed as long as the querying-processes have not finished. The latter is determined from Qt-signals sent by the assignments.

Figure 39

query-results-form



As Figure 39 exemplifies, the query-results are grouped by repository for reasons of clarity and by right-click on an item resembling a file-path the application tries to open that file, if it represents a picture. The same holds true for $S_{caleable}V_{ector}G_{raphics}$ stored in a text-column, so that together with a find or archaeological feature corresponding photographs can be managed. BLOBs are not displayed in the results-form, but have their own mechanism for display, in case they are images. (see chapter *2.4.5.9. BLOBs*). For textual representation of geometries SpatiaLite offers several built-in functions converting geometry to $G_{eography}M_{arkup}L_{anguage}$ or $W_{ell}K_{nown}T_{ext}$.

So in a nutshell the querying mechanism of the desktop-application allows for the user to build complex statements (also nested or containing nested functions) without requiring an understanding of SQL, which is a major concession to the unskilled user's capabilities, and makes parallel interaction with multiple databases possible. While this already covers most archaeological use-cases mainly driven by searching and finding data, the already mentioned query across two repositories accommodates for inter-project data-comparison. Since geometry does not receive special treatment in the result's form, the following chapter describes a simple visualisation-component.

2.4.5.8. Mapping

Because later chapters will describe another mapping option designed into the software-architecture, the mapping mechanism of this chapter is kept rather simple and only intends to display the geometries' relations to each other in 2D space, in order to roughly allow for checking the correctness or spatial distribution. The starting point for mapping is the results-form, where the geometry must be available as $W_{ell}K_{nown}T_{ext}$.

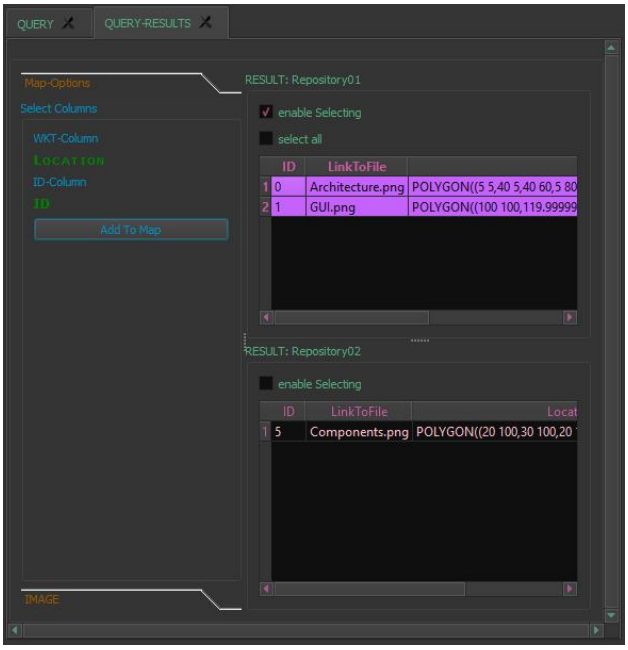


Figure 40

query-results-form

After the column-headers of the results-table were clicked, so as to select the geometries-column as well as an identifier-column - both essential for mapping -, the rows to be illustrated must be marked, before finally the AddToMap-button can be clicked sending the geometries to the mapping-surface, where the spatial information is extracted from the $W_{ell}K_{nown}T_{ext}$ format, in order to create the graphic-items.

Purpose and Preconditions

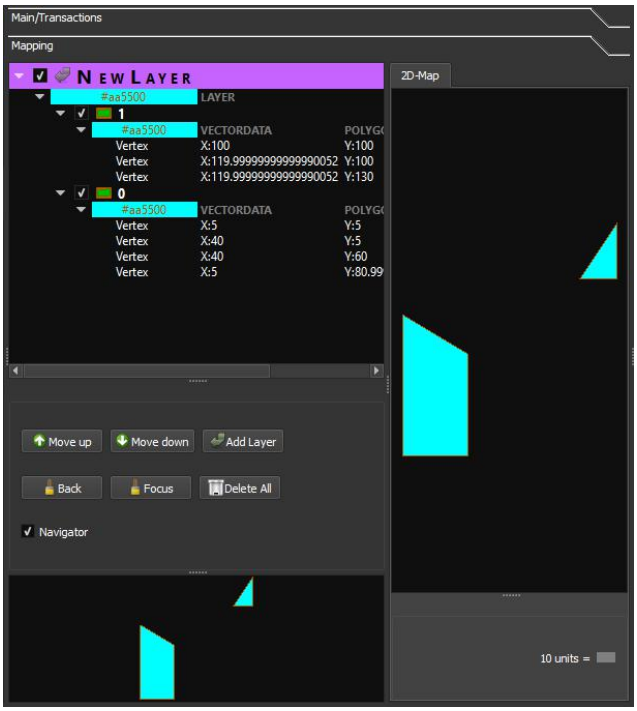


Figure 41

mapping surface featuring map, navigation window and layer-control using Qt's 2D graphics

The mapping device features not only the actual map, a little navigator, a dynamically adapting scale-bar and a list of layers holding the geometries, but also responds to interactions such as panning or zooming on mouse-wheel-events, since according to the user's expectations stipulated by all sorts of web-maps an intuitive handling is aspired. Of course, the layer as well as the single geometry allow for the various visual attributes, such as colour or opacity, to be manipulated via context menu, while furthermore the listed item is interrelated with the mapped geometry thus enabling the user to find the one by selecting the other.

It has to be stressed, however, that the mapping-pane does not intend to be or replace a fully fledged desktop-$G_{eographical}I_{nformation}S_{ystem}$ like QGIS[72] and does not account for reference-systems or projections, because the geometry coming in any case from query-results can easily be projected in the course of the query using SpatiaLite's built-in functions.[73] This guarantees the freedom to map even geometries created on the fly by the query. Although on the one hand there is a Qt-module dealing with maps[74] and on the other hand QGIS libraries operating on the Qt-framework also appear applicable for building a map, both options are unfit to leverage geometry-display here, because Qt's library is not available for C++ (only QML) and QGIS until now relies on the (old) Qt4 libraries thence resulting in incompatibilities. Besides, QGIS' map-visualisation hardly surpasses Qt's default 2D-graphics capabilities exemplified at Figure 41.

Implementation

From a programmer's technical viewpoint even the basic mapping and map-handling of Figure 41 demand more sophisticated coding than non-graphic-related widgets do, not to mention the challenges accompanying multi-threading. As soon as the essential issues on how to represent or manage the spatial features and how to handle user-interaction are solved, getting the geometry onto the map, however, is comparatively straightforward.

The main widget inheriting from QSplitter[75] manages all interplay between the geometries/layers-list, the buttons and the display of graphics. While the layers' list actually is a simple QTreeView[76] operating on a QStandardItemModel[77],

[72] Although QGIS is capable of loading geometry right from the SpatiaLite-database (see for example: http://docs.qgis.org/2.0/de/docs/training_manual/databases/spatialite.html <last visited: 09.03.2017>), it appears that it can only load the full count of a table's entries leading towards performance penalty or program-crash when big-data is concerned. A more scalable approach is thus attempted here.

[73] see: http://doc.qt.io/qt-5/qsplitter.html <last visited: 09.03.2017>.

[74] see: http://doc.qt.io/qt-5/location-maps-cpp.html <last visited: 09.03.2017>.

[75] see: http://doc.qt.io/qt-5/qsplitter.html <last visited: 09.03.2017>.

[76] see: http://doc.qt.io/qt-5/qtreeview.html <last visited: 09.03.2017>.

which in contrast to a widget can be modified form a worker-thread, the maps themselves inherit from QGraphicsView[78], so as to display a QGraphicsScene[79] holding the graphic items.

Figure 42 allows to get an impression on the functionality of the map (-view) itself, while several functional groups become apparent. There are, for example, overridden event-handlers, that together with other class-members take care of panning, zooming as well as dynamically displaying coordinates and a scale-bar. A few class-members are part of coordinating the view with the mini/navigator-window (also inheriting from QGraphicsView), which - since sharing the same scene - requires the map-class to dynamically have a rectangle drawn around the displayed section, in order to give the user a reference of the section shown in the map by displaying aforesaid rectangle in the mini/navigator-window.

Figure 42

class-diagram of the map

Not captured by Figure 42 is the context-menu, that lets the user select a graphical item and change its visual characteristics.

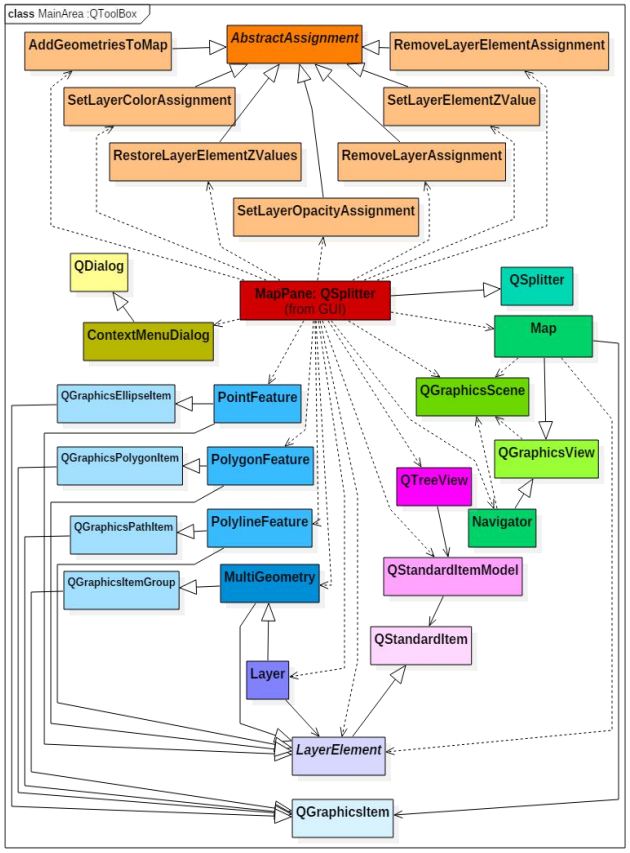


Figure 43

class-diagram of the mapping-pane

As in the scope of this paper the complete implementation of the mapping facility can not be illustrated reasonably as a whole, Figure 43 attempts to grant at least a glimpse of insight into the classes' interplay and exemplify, thus, core principles of the functioning.

In this regard it is to be noted that spatial feature-classes inherit from QGraphicsItem as well as QStandardItem, in order to serve as both at the same time thence eliminating the necessity to correlate layer-list-items with graphic-items. While there are three classes for the basic geometry-types, there is also a container class called MultiGeometry not only uniting multiple spatial-features at once, but also forming the base for the layer-class. Neither QStandardItems nor QGraphicsItemGroups - namely the underlying Qt-Classes - hinder attaching

[77] see: http://doc.qt.io/qt-5/qstandarditemmodel.html <last visited: 09.03.2017>.
[78] see: http://doc.qt.io/qt-5/qgraphicsview.html <last visited: 09.03.2017>.
[79] see: http://doc.qt.io/qt-5/qgraphicsscene.html <last visited: 09.03.2017>.

their kind repetitively as child-items, which means that there can be layers containing multi-geometry comprising simple geometries. With the concept of layers the ordering of geometries' superposition is best geared by the sequence of layers, so that, for instance, the spatial-features of the topmost layer are situated on top of all others. To this end QGraphicsItems store a Z-value simulating the Z-coordinate, which must be updated (in a cascading manner) each time anything in the layers-list is changed. Because with the number of geometries increasing the work-load for updating and similar tasks rises due to nested geometries and list-items, corresponding assignment-classes for concurrent processing are employed (see Figure 43).

In order to sum up: the mapping facility is capable of displaying the spatial-features stored in the databases, while it responds to common user-interaction in terms of navigating over the data, be it on the map itself, be it on the layers/features-list. It is, however, not a $G_{eographical}I_{nformation}S_{ystem}$, because basic spatial tasks are to be handled by the database-system, before the geometry is displayed.

2.4.5.9. BLOBs

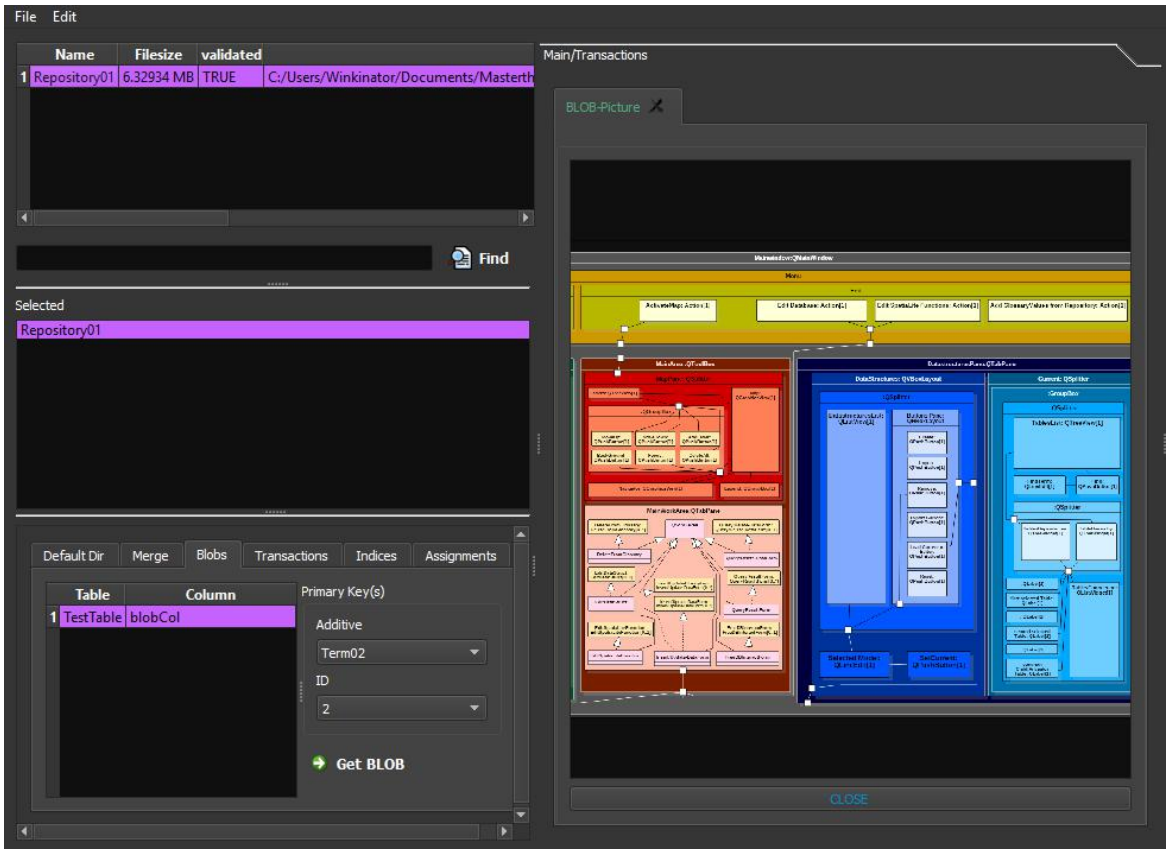


Figure 44

BLOBs-pane (bottom left) with already retrieved BLOB (central right)

Because BLOBs are hardly subject to database-tasks other than storage, their display in case of pictorial character is spun off from the conventional querying procedure. The BLOBs pane (see Figure 44) holds a list of columns containing BLOBs, so that, as soon as a column as well as a repository are selected, the pane lets the user choose from primary-key-column-values, in order to retrieve the according BLOB-picture for display on the main-window. The actual retrieval is done on a worker-thread (for the principles of the mechanism see chapter *2.4.4.1. ThreadManager, Assignment and LockFreeQueue*) employing a byte-array redirected to a file, while a window indicating the program being busy from the current process is displayed.

2.4.5.10. Long Term Assignments

A key-feature of the desktop-application is its capability to run long-term assignments without blocking the user interface or unrelated functionality. Such assignments are, for example, importing a repository, merging repositories or setting an external index (see chapter *2.4.5.12. External Indices*), all of which are likely to run for several minutes or longer. The special aspect here is not only having the work done in a separate thread - in a way discussed in chapter *2.4.4.1. ThreadManager, Assignment and LockFreeQueue* and involving measures for thread-safety - but occupying not more than one thread per assignment, in order not to delay other tasks or generate over-scheduling. During the time of execution the involved repositories are removed from the repositories-list at the GraphicalUserInterface, so as to prevent interference with the databases, and an entry is added to the assignments-list (see Figure 5), both of which modifications are restored at the completion of the long-term tasks. Errors are, of course, reported to the user.

2.4.5.11. Merging Repositories

The ability to merge repositories fulfills the essential need to allow for multiple individuals to operate on the same set of given database-entries during data-collection. So if, for example, there are several teams of archaeologists on an excavation site, each team is handed over a repository-file with glossaries and project-related data pre-set, so that the individual teams conduct their own data-collecting, before it all is assembled into a single project-wide data-store. Additionally, the merging-functionality may also be invoked for reason of reducing data separation over several repository-files or creating temporal data-stores.

In technical terms merging repositories exploits the long-term-assignments-mechanism discussed above (*2.4.5.10. Long Term Assignments*), in order to set up a connection across two separate repositories (like described at 'Indices' in the chapter *3.4.4.4. ConnectionUtility*), that serves as base for SQL-statements of the form "INSERT OR ROLLBACK/FAIL/ABORT/IGNORE/REPLACE INTO main.tableName (columnNames...) SELECT columnNames... FROM repositoryXY.tableName". In doing so, those statements copy the data in accordance with the on-conflict-clause (ROLLBACK etc.) employing database-system-internal mechanisms only. This means - from the user's viewpoint - that beside defining the source- and target-repositories on the corresponding pane (see Figure 45) the on-conflict-mode must be selected, before the process is started.

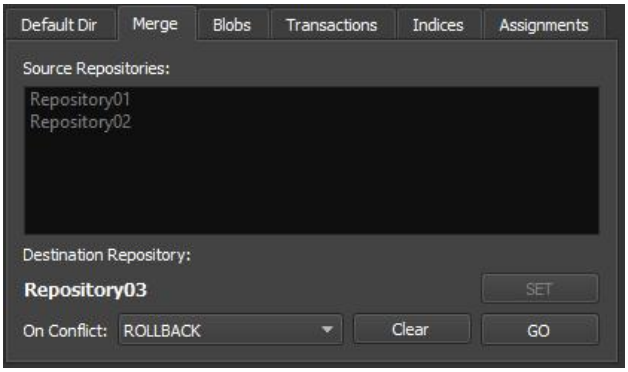


Figure 45

merge-repositories-pane

As usual, all occurring errors are reported to the user in a dialogue.

2.4.5.12. External Indices

Like discussed above (see 'Indices' in chapter *3.4.4.4. ConnectionUtility*) each external index-database is intended to hold entries of multiple repositories, however only in regard of a certain table each. This brings along not only updating external indices in repository-transactions such as inserting, updating or deleting, but requires methods working similar to merging repositories (explained above in chapter *2.4.5.11. Merging Repositories*) for setting or un-setting repositories for indexing, since both undertakings concern copying data from one database to another. Thanks to external indices being internally divided into separate tables resembling repositories removing a repository simply requires deleting a table.

For the concept of this mechanism demands the ability to retrieve information from external indices, the G*raphical*U*ser*I*nterface* has two corresponding options in store for querying.

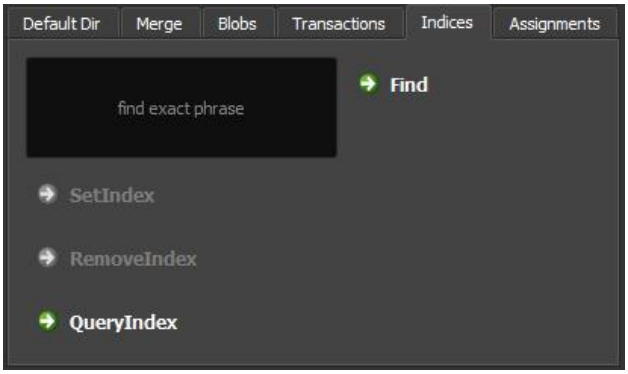


Figure 46

(external-) indices-pane

The one of the two options lets the user find the count of entries matching exactly a specified phrase in a certain column, so as to have the internal SQLite-index speed up the query significantly. While the concerned column and repositories have to be selected in the usual way, the results are shown in a new tab on the main-work-surface and form a table containing repository-names and corresponding entry-counts.

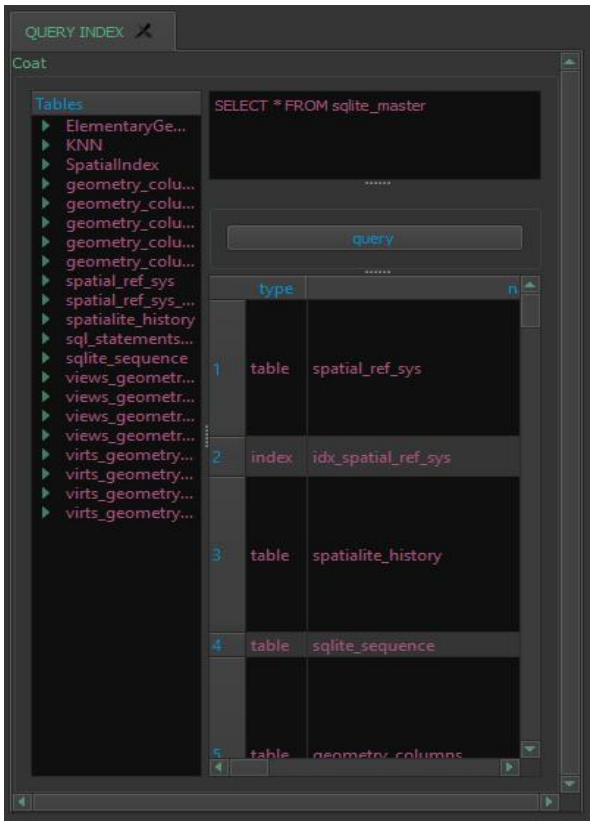


Figure 47

query-external-index-form

The QueryIndex-button (see Figure 46) activates the other query-option by opening a form (see Figure 47), that allows the user to phrase SQL-statements and display the results in a table. It suffice to say that, in doing so, the user is able to conduct data-exploration and -analysis freely in the scope of a table, however across repositories registered for external index. The form is similar to the one discussed in chapter *2.4.5.13. Helper-Tools*, but its functionality is reduced to querying.

Both methods regarding querying external-indices apply the multi-threading mechanism bespoken in chapter *2.4.4.1. ThreadManager, Assignment and LockFreeQueue*, but while the second method (Figure 47) makes the user wait for the completion of the query bridging the time-delay with a modal processing-dialogue, the first approach (Figure 46) works just like the form for outputting repositories' query-results (see *2.4.5.7. Query Results*), for the same custom-widget is employed.

As data-exploration and -analysis usually starts with finding relevant data-sets, the principle of external-indices is, first of all, aimed at helping the user pick out relevant repositories.

2.4.5.13. Helper-Tools

Free Database-InteractionTool

Assuming situations, in which the user seeks to interact with a SpatiaLite-database without being geared by the data-model or the application's restrictions, the edit-menu (see action 'Edit Database' in Figure 48) lets the consumer activate a tab (see Figure 48) allowing for sending statements to a database freely .

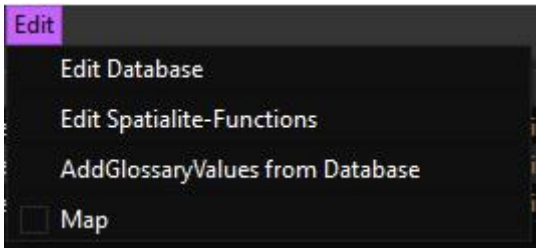


Figure 48

edit-menu

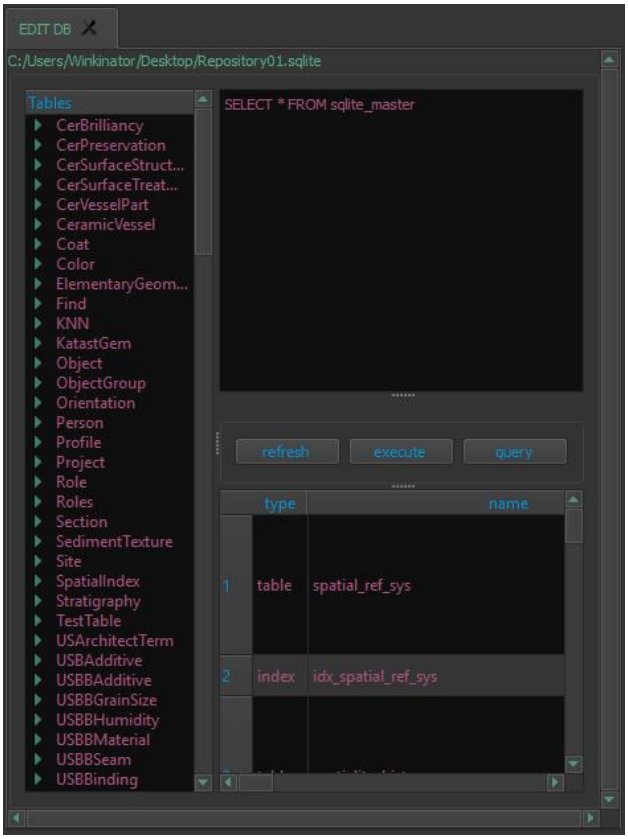


Figure 49

free database-interaction-tool

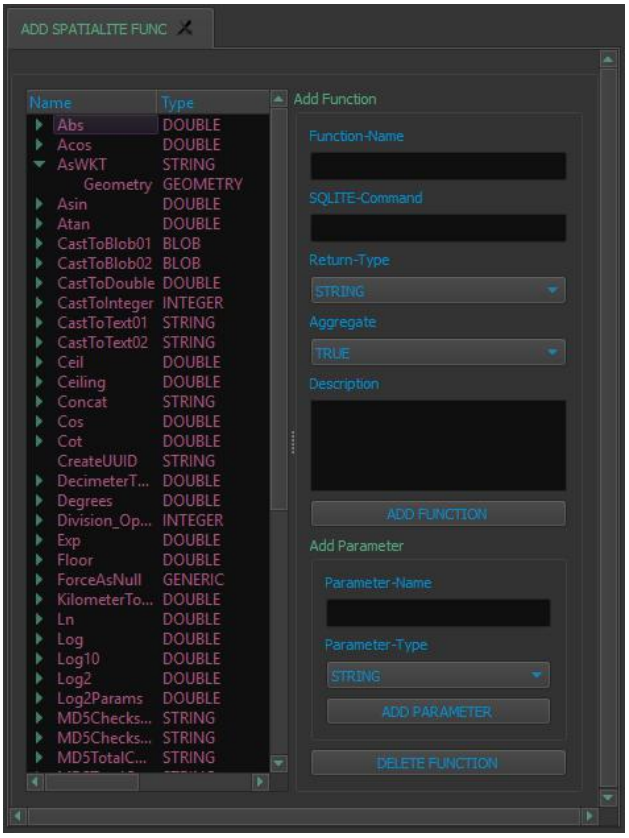


Figure 50

form for editing SQLite/SpatiaLite-functions

This way, a corrupted repository can be repaired by the user manually, so that it is ready for reintegration into the list of registered repositories. A restriction of this tool, however, is its reluctance of connecting to registered repositories or index-files, so as to not undermine the application's authority. Technically the functioning of the tool is straightforward, since it only holds a connection to the database and displays results of statements sent. Additionally a table of contents is shown.

Editing SQLite/SpatiaLite Functions-List

Because the definitions of SQLite/SpatiaLite's built in functions have to find their way into the master-database (see chapter *3.4.4.4. ConnectionUtility*) in an updateable way, so as to have them available for queries (see chapter *2.4.5.6. QueryBuilder-Form*), an input-form (see Figure 50) is provided transferring changes to the corresponding tables of the master-database.

Since the form allows for specifying the function's commands, parameters and corresponding value-types, the SQLite/SpatiaLite-Functions-form is able to guarantee valid input for arguments.

Importing Glossary-Values from Database

Designed as a feature of convenience the application is able to load glossary- values from a database into the glossary-documents provided that no repository is currently registered, a certain glossary-table is selected in the tables-list and the database contains a corresponding table. The underlying mechanism is straightforward and simply consists of having the user select the database in a file-dialogue, connecting to the database, checking the table in the database, querying the values and writing them to the $_{e}X_{tensible}M_{arkup}L_{anguage}$-glossary-file.

2.4.6. Results

In the manner of a case-study the preceding chapter not only unveiled the necessary capabilities of an application striving for leveraging the archaeological recording- and data-management-process, but also discussed a way to attain implementation of the required functionality. Since this is therefore and first of all a software-development-project, the actual results rest in the desktop-program itself representing part of the implemented conceptual solution.

Hence, bearing in mind the goals of this endeavour outlined in the chapter *1.2. General Objectives*, namely constructing an application freeing archaeologists from individually handling similar data and bringing, thus, a gain on data-integration, each of the presented parts of the application has its righteous place fulfilling an essential task, for the various components of the GraphicalUserInterface represent different stages in the work-flow of an archaeologist conducting excavation. It is consequently not just a certain function, but the idea of employing a swarm of database-files (instead of a single database-system) as well as the entire corresponding aggregation of measures, technologies and concepts together with their implementation, that make the intended core benefit of this endeavour, so as to improve archaeological data-handling particularly in spatial terms.

In order to sum up the knowledge gained from building up the desktop application, corresponding explanations must start with the requirements determining the design of the software from the very beginning in many ways: The fact, that the data-model together with the glossaries must be readable by humans and machines - thus be available as structured text documents -, while at the same time being the most basic building-block of the whole system, leads to the necessity to stipulate a specification for those text-documents and build the software up based on those very specifications. So as a first result of this process the most suitable approach is found to be eXtensibleMarkupLanguage-documents for data-model and glossaries, whereas their structure is specified by XSD-schema-documents (see chapters *2.2. Data-Model* and *2.3. Glossaries*) and presumed by the program. Regarding external resources the classic policy of keeping all dependencies in sub-directories of the program is successfully enforced, so as to allow for portability between different computers. For the same reason SpatiaLite is chosen for underlying database system, because among the popular freely available spatial-database-systems it appears to be the most mature of the non-server-like systems, the latter being unfavourable in this endeavour. As a result and in contrast to other database-systems the SQLite-source-code can be built into the application instead of the program being client to a database-server, which leads to the most central insight and - consequently - paradigm of this paper: Given the peculiarities of the field of application and due to archaeological projects being separate entities each finding easily place in SpatiaLite-database-files a file-based non-server-like data-storage mechanism can lead to a very flexible and robust structure however leaving access management to the application. Part of that concept, on the other hand, is not only the requirement to implement various kinds of operations on database-files-management (creating, copying, merging etc.), but to exploit hardware-resources for operations involving multiple database-files in parallel, which is to say that multi-threading is needed, as soon as the application tries to access different databases simultaneously with the intention to compensate for the dispersed data-storage. As a side-effect - but nevertheless expected - the program scales well with the capabilities of the hardware.

In this regard the past chapters demonstrated that by utilizing the Qt-framework together with the C++ programming language a portable, self-sustained and potentially well performing program can be built up capable of not only enabling a technologically unskilled archaeologist to manage excavations' data consistently with a predefined data-model, but also facilitating a simple and robust local data-sharing mechanism relying on local files-exchange for intra-project data-transfer due to the hazard of lacking internet connectivity. Additionally the development-process unveiled the chosen-framework's ability to allow for generating a simple mapping device.

Prospects

Because the application is already comparatively extensive in its core-functionality, the implementation of some rather specific features is left to future attempts, one of which being supplying the implemented map with support for shapefile- and geotiff-import. Since in the course of publishing data the following chapters will introduce other mapping tools and because the application does not intend to be a $G_{eographic}I_{nformation}S_{ystem}$ - as said before -, the desktop-program does not yet take advantage of Qt's support[80] of OpenGL[81], in order to facilitate 3D visualization. Although the latter, too, is probably a beneficial objective for future projects, the website discussed in the following chapters already makes use of a similar technology named WebGL[82] anyway, so that the utilization and benefits of this kind of graphics-library is demonstrated here in any case. Beside that, other endeavours may also attempt to exploit SQLite's full capacity, of which only the core-functionality made it into the application yet.

[80] see: http://doc.qt.io/qt-5/qtgui-index.html <last visited: 15.03.2017>.
[81] see: https://www.opengl.org/ <last visited: 15.03.2017>.
[82] see for example: Anyuru 2012.

2.5. Publishing the Data

Whereas the past chapter presented a solution for on-site archaeological data-management covering merely organizational concerns of an excavation-project, this chapter deals with making the data available outside the project's scope via network (internet).

The basic idea is to host a website entrusted with conveying an impression of the data associated with each data-model, so as to help the remote user decide, which data to obtain. While the resulting data-transfer in this case is achieved simply by downloads as well as WebFeatureService - the latter representing an acknowledged standard after all -, the website itself serves both, the remote and the local user.

Like hinted by Figure 1 and Figure 2, the main components on the server-side are the server itself, a script-interpreter, a script-library for accessing the database-files as well as server-side scripts, while the client is sent an HTML-document holding the basic information together with JavaScript-scripts and style-information codified as CascadingStyleSheet. So irrespective of dynamically assembled documents such as certain download-files and the WebFeatureServic the following pages deal with selecting a suitable server together with a server-side scripting language, before the websites functionality is discussed.

2.5.1. Server-Side-Scripting

Since dynamic websites necessitate some kind of server-side scripting, the most essential question in terms of web-technologies centers upon the scripting-language required to support connections to the SQLite-database-system. Among the various language-bindings of SQLite[83] mainly three languages appear suitable for server-side tasks, namely Ruby, Python and PHP, the latter of which being obviously the only language specialized in server-side scripting [Tatroe and others 2013]. Since PHP not only provides classes for interaction with SQLite as part of its standard-library [Tatroe and others 2013; Owens 2006; http://www.php.net/manual/de/class.sqlite3.php <last visited: 16.03.2017>], but also occupies by far the greatest share of the marked [Tatroe and others 2013; https://w3techs.com/technologies/overview/programming_language/all <last visited: 16.03.2017>] indicating an active community as well as good support, PHP is the best choice for server-side-scripting in the course of this endeavour.

2.5.2. The Server

Fortunately, when it comes to http-servers, one can resort so several free and ready-to-use solutions available via download. Besides simply serving HTML-documents a server is - in this undertaking- required to be applicable on the three main operating systems, as much portable as possible, ready to generate dynamic content (via PHP-scripts) and have good performance. As it appears, the currently most popular http-servers covering the largest part of the market and fulfilling the requirements are Apache-HTTP-Server and NGINX [https://opensource.com/business/16/8/top-5-open-source-web-servers <last visited: 16.03.2017>; https://w3techs.com/technologies/overview/web_server/all <last visited: 16.03.2017>]. Accepted that the applied server is to be recruited from those two options - as this paper aims at exemplifying procedures on prevalent technologies -, in terms of performance and footprint (on the system) [Nedelcu 2013] NGINX is in favour.

Because information on how to have NGINX prepared for serving dynamic websites - respectively setting up CommonGatewayInterface with PHP - is provided elsewhere [see Nedelcu 2013 or https://www.nginx.com/resources/wiki/start/topics/examples/phpfastcgionwindows/ <last visited: 16.03.2017>], the according specifications are not to be repeated here in detail, but it suffice to say that on the one hand the PHP-CGI-executable must be activated listening to the specified port, while on the other hand NGINX must be aware of those very connection-parameters codified in the server's configuration-file.

[83] see for example Owens 2006 or https://en.wikipedia.org/wiki/SQLite <last visited: 16.03.2017>.

2.5.3. Scripts and Documents[84]

By reason that due to portability the servers default directory for websites (html-directory) is not changed, however, still utilised, the following structure of files and directories accomplishes data-publishing on the network and extends, therefore, the desktop-application's directory.

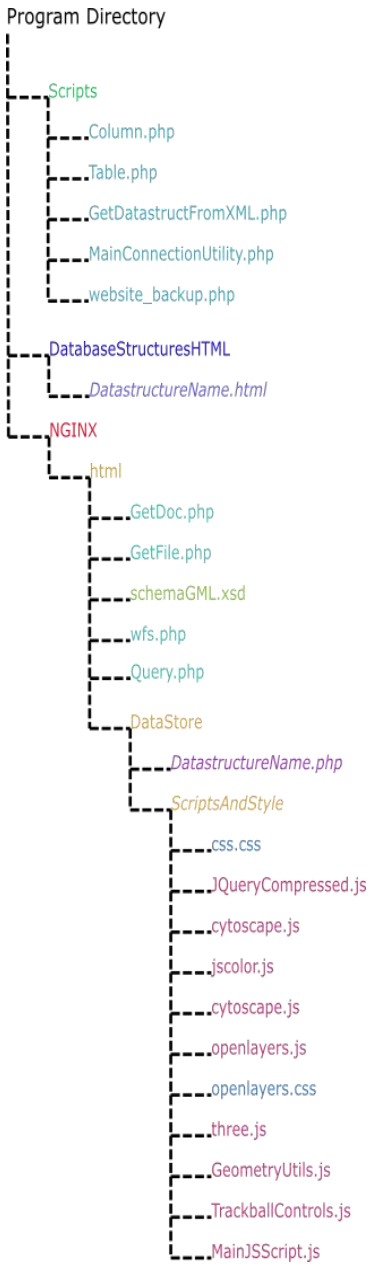


Figure 51

schema of directories and files in the application's directory (extending Figure 6)

In regard of Figure 51 it is important to mention that any file within the html-directory is accessible via network, while the Scripts- and the DatabaseStructure-HTML-directories are not published.

Among the several functional groups of files illustrated in Figure 51 the most essential centers upon *DatastructureName*.PHP and the ScriptsAndStyles-directory, because the php-files carrying the names of data-models (for now substituted by *DatastructureName*) represent the actual websites callable via network, while the ScriptsAndStyles-folder holds the JavaScript[85]-scripts and stylesheets referenced by the websites. This means that by virtue of separate php-files each data-model is granted its own website and address, whereas scripts are shared. Although in doing so redundancy in files comes about, since *DatastructureName.php*-files contain ever the same information, and alternative solutions may likewise be applicable, a simple mechanism is generated, that allows for the desktop-application to update the web-content simply by copying or removing those files embodying the websites, in the context of which the file named website_backup.php serves as source for copying. Anyhow, the Cascading Style Sheet[86] named css.css represents the main source of styling rules responsible for the websites' appearance, whereas the MainJSScript.js-file takes care of all functional aspects, some of which - such as dropdown-menus -, however, are sought to be accomplished by style-rules. The MainJSScript's task is basically to register one big event-handler, that - called, as soon as the website has loaded - furnishes the website with all sorts of features, among which event-handlers prevail. [see Flanagan 2011]

Among the remaining JavaScript-scripts only JQueryCompressed.js must be discussed immediately, since the JQuery-library[87] represents a framework, that lets the scripts interact with the website's elements in a standardized way irrespective of browser-related peculiarities [Flanagan 2011], and is, hence, used all throughout the MainJSScript.js-script.

Beside the website-document two further groups of PHP-scripts executed, of course, on the server-side can be discerned, first of which - while populating the so called Scripts-directory - pursues the intention to manage database-interactions as well as generate html-documents, that while being stored in the DatabaseStructuresHTML-directory represent the data-models as HTML. The other group of PHP-

[84] Because the scripts and documents bespoken in the following chapters are as short as comprehensible and thus rarely require verbal or figurative explanations, they are - unlike the desktop-application's source-code - appended to this paper (see Appendix).

[85] see for example Flanagan 2011.
[86] see for example Nixon 2014.
[87] see https://jquery.com <last visited 19.03.2017>.

scripts resides in the html-folder and takes care of answering asynchronous tasks, more precisely database-queries and downloads.

2.5.4. The Website

Before this chapter will go through the website's different parts (for the PHP/html-code see *[DATASTRUCTURE].php* in Appendix) each representing a kind of theme complex such as querying a repository or mapping, it must be outlined, how the data-model finds its way onto the webpage, since it is neither included statically with the html-code, nor is it to be read dynamically from the $_{e}X_{tensible}M_{arkup}L_{anguage}$-document or the master-database due to performance-considerations. Thus, the PHP-code at the beginning of the website-document examines its own file-name and investigates the DatabaseStructuresHTML-directory for a file with the same name, which will contain data-model-related html-code to import. If no such file is found, the according scripts of the Scripts-directory are called (see *GetDatastructFromXML.php*), in order to generate the required file from the data-model document, before the result is included in the website as an unordered list.

2.5.4.1. Data-Model

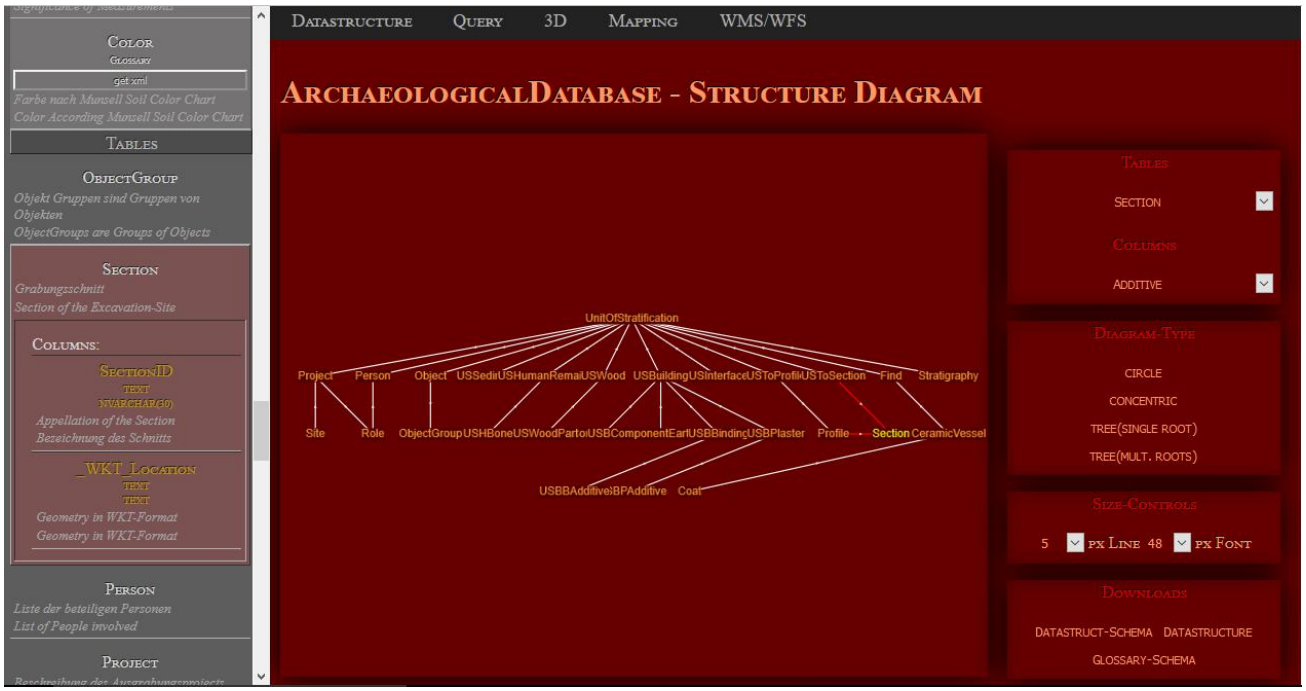


Figure 52

first section of the website presenting all necessary information on the data-model

The fixed pane on the left of Figure 52 portrays all data-model-related information on tables and columns presented as nested (but folding) lists, that not only allow for navigating along foreign-key-connections and synchronises with the diagram as well as the dropdown-menu of tables at the right-hand-side, but also lets the user download the various glossaries into the browser. The ordering of tables, thereby, follows the same paradigm as outlined in chapter *2.4.5.2. The current Data-Model*. While at the bottom right of the data-model-area the XML-schema-definitions as well as the data-structure can also be loaded into the browser (similar to Figure 53), the rest of the widgets deal with the diagram of the data-model. This very graph is the result of the cytoscope.js-graph-library released under the permissive open source license (MIT)[88] and is implemented here, so as to appear in four different forms, all of which only display non-glossary-tables together with their foreign-key-interconnections. Although the tables in the graph

[88] see http://www.js.cytoscape.org <last visited 18.03.2017>.

can be dragged around freely and selected, so as to cause the data-model-pane and the tables-dropdown-menu to focus the corresponding table, the view can be modified according to the four named appearances by clicking the buttons on the right.

In technical terms the pane carrying the graph is a general-purpose html-container, until the MainJSScript (see *MainJSScript.js*) - the source of all dynamic behaviour and event-handlers - creates the rather complex cytoscape-object together with all its related attributes, one of which being a reference to the html-container. The ordering-system of elements in the graph's predefined layouts is determined by the arrangement and count of table-connections, respectively foreign-keys.

The functionality of the mentioned downloads (see Figure 53), on the other hand, is rooted in the script GetDoc.php (see *GetDoc.php*), which is called by the website asynchronously [for AJAX see for example: Hollosi 2012] and simply reads the text-files' content, before setting the Content-Type-header to 'text/xml' and sending out the http-response containing the read content.

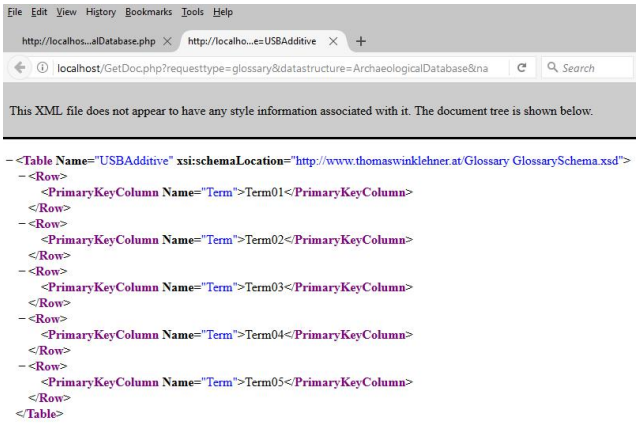


Figure 53

glossary loaded into the browser

So in a nutshell, the first section of the website takes all measures to convey an impression of the data-model in a preferably interactive way importing only technology, that is free to use. Among the latter the graph accomplished by cytoscope.js resembles a a first spatial element not depicting geometry, but a network. Besides, given the available downloads a remote user can obtain all information necessary to reproduce the exact same data-model via ready to use textual documents.

2.5.4.2. The Query-Builder

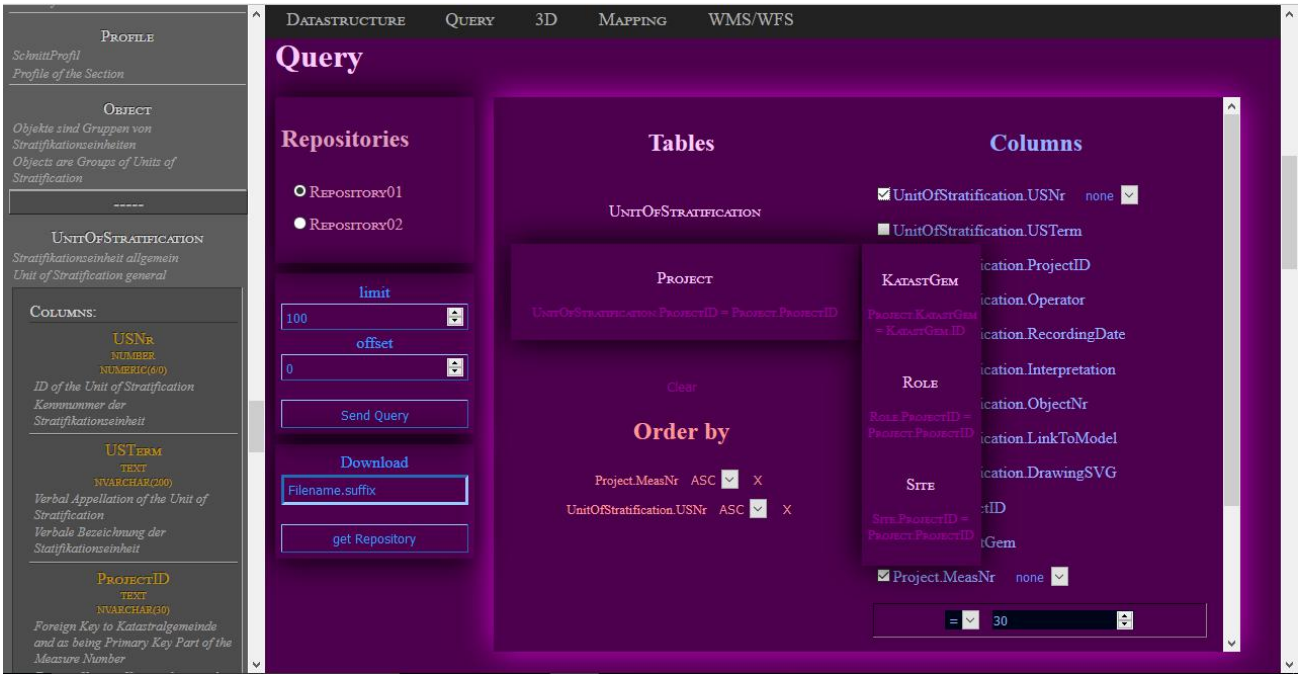


Figure 54

the query-builder-interface

The query-builder - although simple - fulfills nevertheless a task similar to the desktop-application's query-mechanism, namely enabling the user to browse through the repositories' data without requiring an understanding of SQL.

The query begins with the table currently selected in the tables' dropdown-menu (see Figure 52), whereas further tables can only be added by clicking the chosen table's sub-menu-items representing related tables (see Figure 54). This way, a chain of interconnected tables is produced taking care of adding each table to the query only once. As a result, the query-builder offers only the columns originating from the chosen tables, so as to have them selected by the user, be it for normal query, be it for aggregate function or as part of a simple condition. Related ORDER BY- or GROUP BY-clauses are filled automatically, in order to have the user only decide, which columns to exclude from ordering or grouping again.

Because in its intention to be as dynamic as interacted with intuitively the query-builder watches over the user's input, it completely eliminates the risk of assembling a syntactically invalid SQL-query. The information on the data-model is gathered from the html-elements forming the fixed pane at the left and made into a dynamic form applying JavaScript. Aside from selecting the repository sought to query and specifying the limit as well as offset of the resulting datasets, the pane for downloads is to be noticed in Figure 54, since it enables the user to download either a copy of the repository (when no file-name is specified) or download a certain file referenced by the repository. The second option is only then successful, if the repository is situated in the default-repository and has the requested file stored in its external files-directory.

Before the following chapter will finally outline the result of clicking the Send-Query-button, the way the query-builder works is to be discussed. Since preparing PHP for loading extensions to SQLite turned out to be a cumbersome procedure - whilst the spatial functionality of SpatiaLite is actually not needed for simple data-requests and may even lead to performance-penalty -, an alternative solution is established requiring the desktop-application to install two further columns with each spatial-column. Those two other fields hold representations of the according geometry in the form of $W_{ell}K_{nown}T_{ext}$ as well as $G_{eography}M_{arkup}L_{anguage}$ and are updated by triggers, so as to jump one process in the course of a query and output the geometry's textual representation directly. The website respects that fact in a way, as it offers only the column's textual-version for query. As far as the actual process of querying is concerned, the according event-handler triggered by the Send-Query-button collects the information and assembles it into a http-request utilizing the POST-Method[89], before the request is sent asynchronously to the server. Another event-handler (see *MainJSScript.js*) is, of course, registered, so as to receive the response generated by the script Query.php (see *Query.php*), that - in a nutshell - checks the transferred SQL-parts via regular-expressions, seeks out the file-path to the repository from the master-database, connects to and queries the latter, before the results are returned in the form of html.

Another asynchronous task is the files-download involving the script GetFile.php (see *GetFile.php*), which basically finds the requested file and reads it to the output, in which process setting the necessary response-headers is crucial, since they make the browser initiate its download-proceeding as shown in Figure 55.

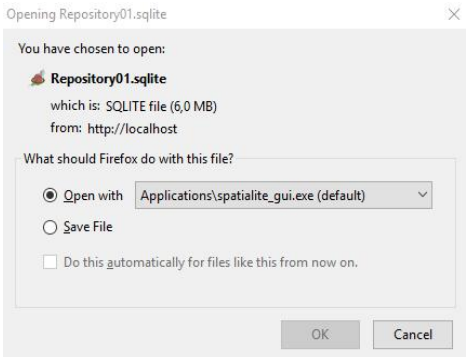


Figure 55

download-dialog

So all in all, the idea behind the Query-area is to equip the user with a rudimentary, but easy to use mechanism to retrieve data from repositories, in order to convey a rough impression of the repositories' content. The functionality is not exhaustive, since advanced tasks are supposed to be executed in full-blown $G_{eographic}I_{nformation}S_{ystems}$ after obtaining the data via $W_{eb}F_{eature}S_{ervice}$ or files-

[89] see for example Flanagan 2011.

download (the latter being another advantage of file-based database-systems). The reason for not offering a shortcut to retrieving a column's entire data at once - so as to populate a results-table or map with the complete amount of data in a simpler manner, is the fact that no assumptions are made on the size of datasets, but - in order to be on the safe side - data-retrieval is portioned. The website, by the way, is surely contrived more openly than recommended for practical use - with the latter probably including user-accounts -, because it is this paper's general objective to illustrate ways, not to restrict chances.

2.5.4.3. Query-Results

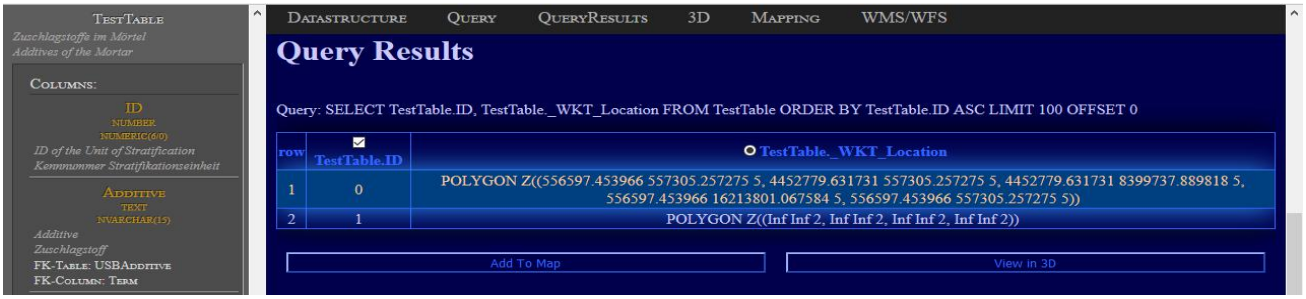


Figure 56

query-results-area

The Query-Results-section primarily displays the SQL-statement together with the query-results laid out as a table. In case the data bears alphanumeric-values only, nothing more happens. If, however, a column is known to contain geometry in the form of $W_{ell}K_{nown}T_{ext}$, radio-buttons let choose between the (potentially several) $W_{ell}K_{nown}T_{ext}$-columns and check-boxes mark data-columns, while at the same time rows become selectable - finally allowing push-buttons to start 2D- and 3D-mapping. All those measures pave the way for map-display and, thus, aim at selecting features together with descriptive attributes, that will serve as labels.

While the dynamic functionality, again, comes from the client-side MainJSScript.js registering the corresponding event-handlers, the major part of the Query-Results-section's content comes from the server-side script named Query.php and is delivered as ready-to-use html-code.

2.5.4.4. 2D Mapping

The website's mapping device represents one of the endpoints of this whole endeavour as far as spatial data is concerned. As the preceding chapter already hinted, the prime source of map-features is the query-results-table guaranteeing for only actually wanted data-entries to be mapped and hardware-resources to be spared.

Like any major $G_{eographic}I_{nformation}S_{ystem}$ the geometry is organized in layers, based on which labelling and features' appearance are managed, while the layers' ordering, thereby, affects the geometries' overlay-behaviour. Assuming that the user's browser is connected to the internet, OpenStreetMap is installed as basemap, so as to provide a spatial-reference from the very beginning. The few controls accompanying the map are rather self-explanatory in the context of aspects already bespoken, except for two forms facilitating basic $W_{eb}M_{ap}S_{ervice}$- and $W_{eb}F_{eature}S_{ervice}$- import (see Figure 58).

The map's underlying functionality is rooted in the opensource openlayers3-library[90] imported as JavaScript-script providing any function needed for mapping on websites. The default distribution, however, comes with the restriction that only two spatial-reference-systems are supported by default,[91] one of which being the common EPSG:3857. Instead of trying to install support for further reference systems via, for example proj4.js[92], the limitation is sought to be turned into a virtue.

[90] see https://openlayers.org <last visited: 20.03.2017>.
[91] others, however, can be specified.
[92] see http://www.proj4js.org <last visited: 20.03.2017>.

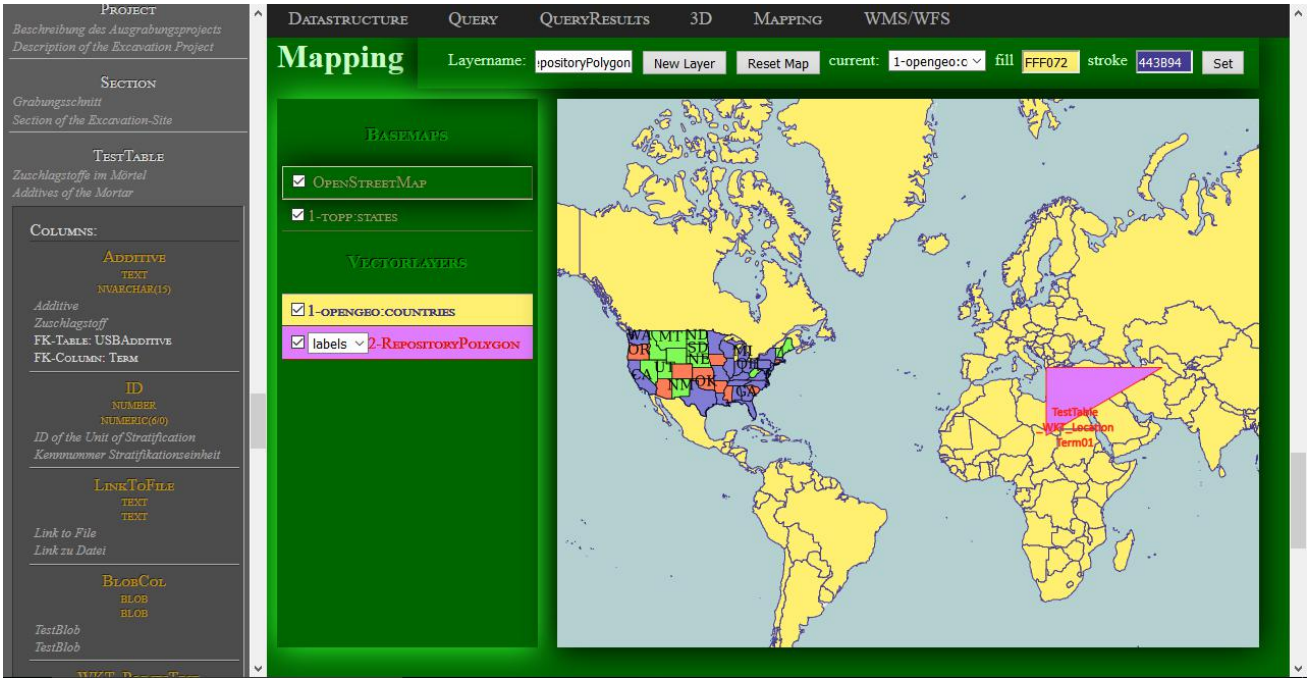


Figure 57

the mapping section displaying a WMS-, WFS- and stored geometry

This very virtue manifests in the fact that, since geometry is retrieved from repositories in the preconceived textual form of $W_{ell}K_{nown}T_{ext}$ with its reference system already established, the user is freed from the burden to match reference systems of repositories and the map, which in turn presupposes, of course, that the database-triggers updating the geometries' textual columns employ EPSG:3857, as they actually do.

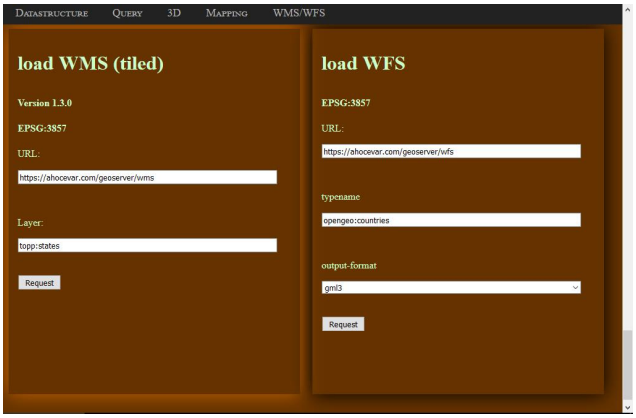


Figure 58

WMS/WFS import-forms

Because EPSG:3857 appears to be the most common reference system anyway, there should hardly ever be a problem importing data via aforementioned $W_{eb}M_{ap}S_{ervice}$- and $W_{eb}F_{eature}S_{ervice}$, as tests revealed, while bearing the advantage of the user being absolutely bypassed the entire issue of reference-systems. It has to be stressed that, while virtually all map-related functionality originates from openlayers, the applied mapping-framework does not rule the market without rivals, since there is, for example, leaflet[93], which obviously follows a more modular approach entailing, though, its more complicated application.

Although the implementation of the website's mapping-capability does not harvest all functions offered by openlayers, the main requirements are met allowing finally for displaying the repositories' geometries together with whatever spatial data is loaded via WMS/WFS. So together with the querying-mechanism the mapping stands for a utility to further investigate the data and help the user discern data to obtain.

[93] see http://www.leafletjs.com <last visited: 20.03.2017>.

2.5.4.5. 3D Visualization

In terms of archaeological excavations two dimensional mapping is not enough, as special meaning rests in the vertical stratification of layers. Thence, the website extends the capabilities of the desktop-application towards 3D-visualization, so that the archaeologist is able to explore the spatial features in 3D-space (see Figure 59).

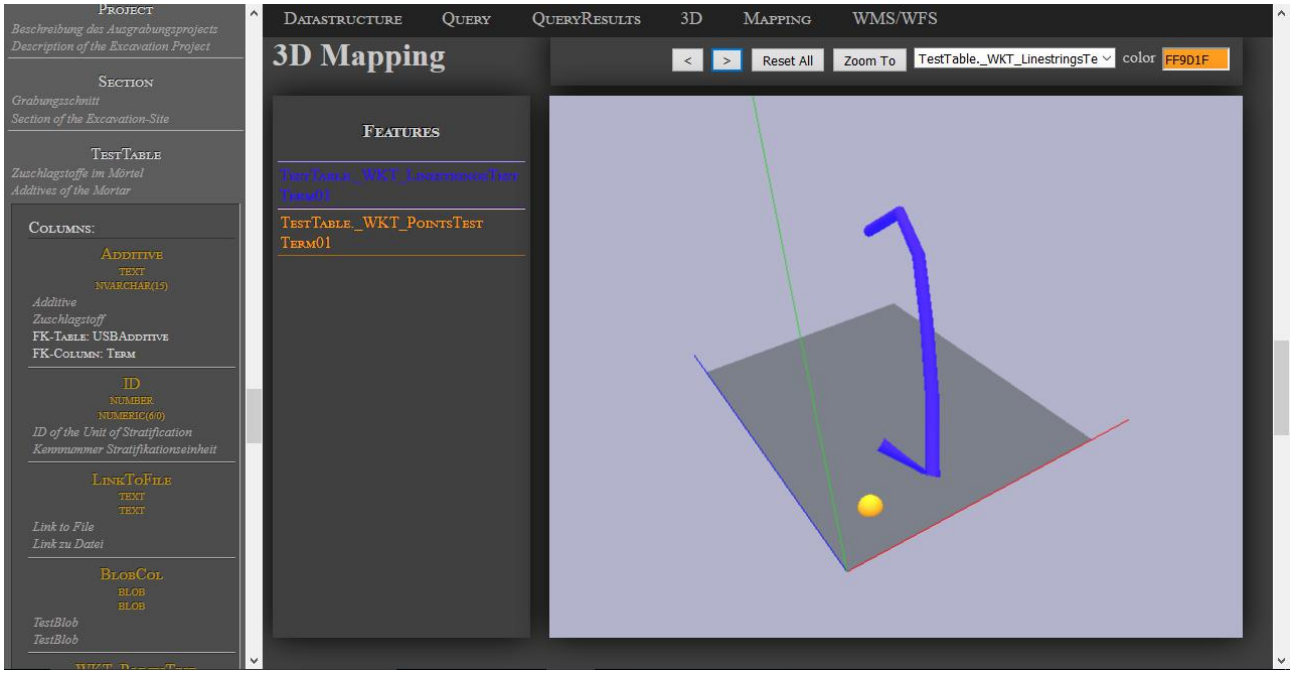


Figure 59

3D-visualization-pane

Like 2D-mapping the starting point for its 3D-counterpart is the query-results-area, from where the spatial data is harvested in the form of $W_{ell}K_{nown}T_{ext}$. In order to reduce memory consumption and computational expense, the query-results can only be added one by one contributing both, geometry and identifier. As the control-panel illustrates, the various geometries can be zoomed to, enlarged and scaled down, while the colour is determined, when the geometry is added. The mouse-buttons and -movements provide the functionality to navigate through the scene intuitively thence giving the archaeologist the chance to explore the recorded stratification in all its spatial extent. The axes of the coordinate system together with the ground-plane adapt automatically with geometries added. The actual functionality is contributed by a javascript-framework named three.js[94], that while being a loadable script represents the interface to WebGL[95], a technology for sophisticated graphics-display on websites. The core object most essential in the process is the so called scene-object requiring first of all an html-container, a renderer, lights and a camera [see ThreeJS 2013 and *MainJSScript.js*], so as to output any graphic at all. Given the necessity of illustrating archaeological stratification by virtue of threejs's range of available geometry-objects[96] point-features are best represented as spheres, linestring- and polygon-features as tubes, so that unlike with simple paths scaling the size (respectively diameter) can be accomplished. Flat surfaces, however, are hardly an option, since archaeological layers usually extend in 3D-space irregularly. The MainJSScript-script (see *MainJSScript.js*) illustrates the steps from extracting $W_{ell}K_{nown}T_{ext}$'s coordinates up to generating the

[94] see https://threejs.org/ <last visited 24.03.2017> or ThreeJS 2013.

[95] see: Matsuda – Lea 2013 or Parisi 2014.

[96] see https://threejs.org/ <last visited 24.03.2017>.

finally shown mesh constituted by the connected vertices' geometry combined with a material. Modifications on the scene - be it adding a feature or resizing a geometry - usually involve redrawing the affected elements and, thus, come at a cost. For this very same reason the applied materials are kept simple, as they do not convey additional information anyway and would only impair rendering performance. Rendering is also the point, where the scene's controls deploy their power exerted by an additional script named TrackballControls.js[97].

So as to sum up the significance of the 3D-visualizer, its quality as supplement to the 2D-mapping device has to be stressed, because, while the latter primarily puts geometry in relation to a reference system, the 3D-scene allows for examining the spatial correlation of features to each other.

2.5.5. The WebFeatureService

Although the capability for downloading repositories already provides full access to all data, another mechanism complying with a standard is sought to be implemented. Like outlined in chapter *1.2.3. Conceptual Solution* this very standard is WebFeatureService and can generally be established employing open-source software such as Geoserver[98] or MapServer[99]. Although browsing akin services on the internet conveys the impression of Geoserver being the most popular solution - probably due to its GraphicalUserInterface -, it bears the disadvantage of requiring JavaRuntimeEnvironment and a container such as Apache Tomcat or Glassfish [Iacovella –Youngblood 2013] consequently hampering portability. MapServer, on the other hand, apparently tending to be applied with Apache-server is a platform without user-interface, that runs natively on the the operating system and specifies its services in so called map-documents [http://www.ms4w.com <last visited 29.03.2017>], that by virtue of their textual nature can be manipulated even by scripts or custom-applications.

So provided that all MapServer-related resources (binaries) comply with the operating system, the necessary modifications entailing the transfer of the component between machines are mostly limited to setting the correct file-paths in map-documents and the server's configuration-files as well as specifying the port Apache is listening to. Although it is beyond the scope of this paper, it is to be stressed that those very adaptations could be enforced applying a self-made script.

So given the aspects mentioned above combined with this endeavour focusing on the windows 10 operating system the exemplified solution to implementing the WebFeatureService is, first of all, installing the Apache-Mapserver-bundle available for download at http://www.ms4w.com. Even though Apache-server is configured for listening to the free port 8080, it is made inaccessible from the network due to modifications set in its configuration file, which not only make it listen exclusively to port 127.0.0.1:80 (the local machine), but also restricts directory-access to the very same local IP-address according to the pattern presented at https://httpd.apache.org/docs/2.4/howto/access.html <last visited 29.03.2017>. After having made Apache a purely local server, the website's host, the NGINX-server, is modified, so as to function as a proxy-server redirecting requests to the local Apache-server. In doing so, not only the service-related capabilities are capsuled at the Apache-Mapserver-bundle thus supporting the software-architecture's modular pattern, but in the process of redirecting request parameters can be modified, as illustrated by the following lines taken from NGINX's configuration-file:

```
server {
 Listen              90;
 server_name         apache;
 root                WFS;
 location ~ /(?<model>.*)/WFS {
     proxy_pass          http://127.0.0.1:8080/cgi-bin/map
     serv.exe?map=$document_root/$model.map&$args;
 }
}
```

[97] see https://threejs.org/examples/js/controls/TrackballControls.js <last visited: 25.03.2017>.
[98] see Iacovella –Youngblood 2013, Henderson 2014 and http://www.geoserver.org <last visited 29.03.2017>.
[99] http://www.mapserver.org <last visited 29.03.2017> and http://www.ms4w.com <last visited 29.03.2017>.

Those lines[100] define, first of all, that only requests from port 90 are redirected and that the root-directory for referenced files is WFS, a subdirectory in the NGINX folder. The WFS-directory is the place, where aforementioned map-documents describing the service reside. Since it is defined as root and settled relative to the primary-server's directory, it can be referenced via the variable $document_root. The phrase following 'location' represents the filter for the parts of the URI tailing the server- and port-specification, so that only requests in the form of 'http://ipAddress:port/datamodel/wfs' are handled. The data-model's name is stored in a variable, in order to be used in the query-string sent to the Apache-server. The query-string of the original request is appended to the modified URI via the variable &args, as visible in the section starting with 'proxy_pass' representing the proxy-target. As illustrated by the proxy_pass-directive, MapServer requires not only navigating to the mapserv.exe-file, but also a parameter hinting the input-map-file. Hence, the proxy-mechanism hides the MapServer-specific input and leaves only the WebFeatrueService-related parameters for the user to set.

Like already implied by the foregoing paragraphs each data-model possesses its own WebFeatureService specified in a separate map-document stored in the NGINX-server's subdirectory named WFS.

The following lines exemplify a basic example of such map-document[101] representing a WebFeatureService-implementation, that corresponds with the data-model outlined above (see *2.2. Data-Model*) and simulates the existence of a repository named Repository01:

```
MAP
 NAME "WFS_server"
 STATUS ON

WEB
 METADATA
  "wfs_title"  "ArchaeologicalDatabase"
  "wfs_onlineresource"  "http://localhost:90/
      WFSTestModel/WFS"
  "wfs_srs"  "EPSG:3857"
  "wfs_abstract"  "Database for arch. Data."
  "wfs_enable_request" "*"
  "ows_keywordlist"  ""
  "ows_fees"  ""
  "ows_accessconstraints"  ""
  "ows_contactorganization"  ""
  "ows_contactperson"  ""
  "ows_service_onlineresource"  ""
  "ows_contactposition"  ""
  "ows_contactvoicetelephone"  ""
  "ows_contactfacsimiletelephone"  ""
  "ows_address"  ""
  "ows_city"  ""
  "ows_stateorprovince"  ""
  "ows_postcode"  ""
  "ows_country"  ""
  "ows_contactelectronicmailaddress"  ""
  "ows_hoursofservice"  ""
  "ows_contactinstructions"  ""
  "ows_role"  ""
 END
END

LAYER
 NAME Repostory01_Find_Location
 TYPE POINT
 CONNECTIONTYPE OGR
 CONNECTION "path/to/Repository01.sqlite"
 DATA "Select ID, Transform(Location,3857) from
                                        Find"
 EXTENT -20026376.39 -20048966 20026376.39
      20048966
 PROJECTION
     "init=epsg:3857"
 END
 STATUS ON
 METADATA
      wfs_title      "Location of Find"
      wfs_featureid  "ID"
      gml_include_items "all"
 END
END


LAYER
 NAME Repostory01_Section_Location
 TYPE POLYGON
 CONNECTIONTYPE OGR
 CONNECTION "path/to/Repository01.sqlite"
 DATA "Select SectionID, Transform(Location,
                              3857) from Section"
 EXTENT -20026376.39 -20048966 20026376.39
      20048966
 PROJECTION
     "init=epsg:3857"
 END
 STATUS ON
 METADATA
      wfs_title      "Location of Section"
      wfs_featureid  "SectionID"
```

[100] for information in how to configure NGINX, see for example: http://nginx.org/en/docs/beginners_guide.html <last visited: 29.03.2017>.

[101] for information on how to structure map-documents see: http://www.mapserver.org/ogc/wfs_server.html <last visited: 29.03.2017> and http://www.mapserver.org/input/vector/sqlite.html <last visited: 29.03.2017>.
On all the available parameters see: http://www.mapserver.org/documentation.html <last visited: 31.03.2017>.

```
      gml_include_items "all"
 END
END

LAYER
 NAME Repostory01_Profile_Location
 TYPE LINE
 CONNECTIONTYPE OGR
 CONNECTION "path/to/Repository01.sqlite"
 DATA "Select ProfileID || \"_" || SectionID AS
     ID,Transform(Location, 3857) from Profile"
 EXTENT -20026376.39 -20048966 20026376.39
      20048966
 PROJECTION
     "init=epsg:3857"
 END
 STATUS ON
 METADATA
      wfs_title      "Location of Profile"
      wfs_featureid  "ID"
      gml_include_items "all"
 END
END

LAYER
 NAME Repostory01_Site_Location
 TYPE POLYGON
 CONNECTIONTYPE OGR
 CONNECTION "path/to/Repository01.sqlite"
 DATA "Select SiteID, Transform(Location,
                      3857) from Site"
 EXTENT -20026376.39 -20048966 20026376.39
      20048966
 PROJECTION
     "init=epsg:3857"
 END
 STATUS ON
 METADATA
      wfs_title      "Location of Site"
      wfs_featureid  "SiteID"
      gml_include_items "all"
 END
END

LAYER
 NAME Repostory01_Building_Location
 TYPE POLYGON
 CONNECTIONTYPE OGR
 CONNECTION "path/to/Repository01.sqlite"
 DATA "Select USNr, Transform(Location, 3857)
                                 from USBuilding"
 EXTENT -20026376.39 -20048966 20026376.39
      20048966
 PROJECTION
     "init=epsg:3857"
 END
 STATUS ON
 METADATA
      wfs_title      "Location of Building"
      wfs_featureid  "USNr"
      gml_include_items "all"
 END
END

LAYER
 NAME Repostory01_Building_TopSurface
 TYPE POINT
 CONNECTIONTYPE OGR
 CONNECTION "path/to/Repository01.sqlite"
 DATA "Select USNr, Transform(UpSurface, 3857)
                                 from USBuilding"
 EXTENT -20026376.39 -20048966 20026376.39
      20048966
 PROJECTION
     "init=epsg:3857"
 END
 STATUS ON
 METADATA
      wfs_title      "Top-Surface of Building"
      wfs_featureid  "USNr"
      gml_include_items "all"
 END
END

LAYER
 NAME Repostory01_Building_BottomSurface
 TYPE POINT
 CONNECTIONTYPE OGR
 CONNECTION "path/to/Repository01.sqlite"
 DATA "Select USNr, Transform(DownSurface, 3857)
                                 from USBuilding"
 EXTENT -20026376.39 -20048966 20026376.39
      20048966
 PROJECTION
     "init=epsg:3857"
 END
 STATUS ON
 METADATA
      wfs_title    "Bottom-Surface of Building"
      wfs_featureid  "USNr"
      gml_include_items "all"
 END
END

LAYER
 NAME Repostory01_HumanRemains_Location
 TYPE POLYGON
 CONNECTIONTYPE OGR
 CONNECTION "path/to/Repository01.sqlite"
 DATA "Select Individual, Transform(Location,
                     3857) from USHumanRemains"
 EXTENT -20026376.39 -20048966 20026376.39
      20048966
 PROJECTION
     "init=epsg:3857"
 END
 STATUS ON
 METADATA
      wfs_title      "Location of HumanRemains"
      wfs_featureid  "Individual"
      gml_include_items "all"
 END
END

LAYER
 NAME Repostory01_Interface_Location
 TYPE POLYGON
 CONNECTIONTYPE OGR
 CONNECTION "path/to/Repository01.sqlite"
 DATA "Select USNr, Transform(Location,
                      3857) from USInterface"
 EXTENT -20026376.39 -20048966 20026376.39
      20048966
 PROJECTION
```

```
     "init=epsg:3857"
 END
 STATUS ON
 METADATA
      wfs_title      "Location of Interface"
      wfs_featureid  "USNr"
      gml_include_items "all"
 END
END

LAYER
 NAME Repostory01_Interface_Surface
 TYPE POINT
 CONNECTIONTYPE OGR
 CONNECTION "path/to/Repository01.sqlite"
 DATA "Select USNr, Transform(Surface,3857)
                              from USInterface"
 EXTENT -20026376.39 -20048966 20026376.39
      20048966
 PROJECTION
     "init=epsg:3857"
 END
 STATUS ON
 METADATA
      wfs_title      "Surface of Interface"
      wfs_featureid  "USNr"
      gml_include_items "all"
 END
END

LAYER
 NAME Repostory01_Sediment_Location
 TYPE POLYGON
 CONNECTIONTYPE OGR
 CONNECTION "path/to/Repository01.sqlite"
 DATA "Select USNr, Transform(Location, 3857)
                                from USSediment"
 EXTENT -20026376.39 -20048966 20026376.39
      20048966
 PROJECTION
     "init=epsg:3857"
 END
 STATUS ON
 METADATA
      wfs_title      "Location of Sediment"
      wfs_featureid  "USNr"
      gml_include_items "all"
 END
END

LAYER
 NAME Repostory01_Sediment_TopSurface
 TYPE POINT
 CONNECTIONTYPE OGR
 CONNECTION "path/to/Repository01.sqlite"
 DATA "Select USNr, Transform(UpSurface, 3857)
                                from USSediment"
 EXTENT -20026376.39 -20048966 20026376.39
      20048966
 PROJECTION
     "init=epsg:3857"
 END
 STATUS ON
 METADATA
      wfs_title      "Top-Surface of Sediment"
      wfs_featureid  "USNr"
      gml_include_items "all"
 END
END

LAYER
 NAME Repostory01_Sediment_BottomSurface
 TYPE POINT
 CONNECTIONTYPE OGR
 CONNECTION "path/to/Repository01.sqlite"
 DATA "Select USNr, Transform(DownSurface,3857)
                                from USSediment"
 EXTENT -20026376.39 -20048966 20026376.39
      20048966
 PROJECTION
     "init=epsg:3857"
 END
 STATUS ON
 METADATA
      wfs_title    "Bottom-Surface of Sediment"
      wfs_featureid  "USNr"
      gml_include_items "all"
 END
END

LAYER
 NAME Repostory01_Wood_Location
 TYPE POLYGON
 CONNECTIONTYPE OGR
 CONNECTION "path/to/Repository01.sqlite"
 DATA "Select USNr, Transform(Location, 3857)
                                from USWood"
 EXTENT -20026376.39 -20048966 20026376.39
      20048966
 PROJECTION
     "init=epsg:3857"
 END
 STATUS ON
 METADATA
      wfs_title      "Location of Wood"
      wfs_featureid  "USNr"
      gml_include_items "all"
 END
END

LAYER
 NAME Repostory01_Wood_TopSurface
 TYPE POINT
 CONNECTIONTYPE OGR
 CONNECTION "path/to/Repository01.sqlite"
 DATA "Select USNr, Transform(UpSurface, 3857)
                                from USWood"
 EXTENT -20026376.39 -20048966 20026376.39
      20048966
 PROJECTION
     "init=epsg:3857"
 END
 STATUS ON
 METADATA
      wfs_title      "Top-Surface of Wood"
      wfs_featureid  "USNr"
      gml_include_items "all"
 END
END

LAYER
 NAME Repostory01_Wood_BottomSurface
 TYPE POINT
 CONNECTIONTYPE OGR
```

```
  CONNECTION "path/to/Repository01.sqlite"
  DATA "Select USNr, Transform(DownSurface,
                          3857) from USWood"
  EXTENT -20026376.39 -20048966 20026376.39
                            20048966
  PROJECTION
      "init=epsg:3857"
  END
  STATUS ON
  METADATA
        wfs_title     "Bottom-Surface of Wood"
        wfs_featureid  "USNr"
        gml_include_items "all"
  END
 END
END
```

The preceding lines exemplify the typical nested structure of a map-document featuring a root-element called 'MAP'. The main information, that follows, is $W_{eb}F_{eature}S_{ervice}$-related metadata (enclosed by the element 'WEB'), before the actual feature-layers signalled by the term 'LAYER' are specified. Since the single items are rather self-explanatory, only the involved transformation of coordinates in the course of acquiring the data via SQL is to be highlighted, because - as it turned out - no automatic transformation in response to request parameters takes place in this very basic map-outlay otherwise. Although the latter only regards the absolute minimum set of specification, the aimed for goal is achieved nevertheless having demonstrated a comparatively simple way to combine the preceding data-management-system with a $W_{eb}F_{eature}S_{ervice}$ entrusted with data-publishing.

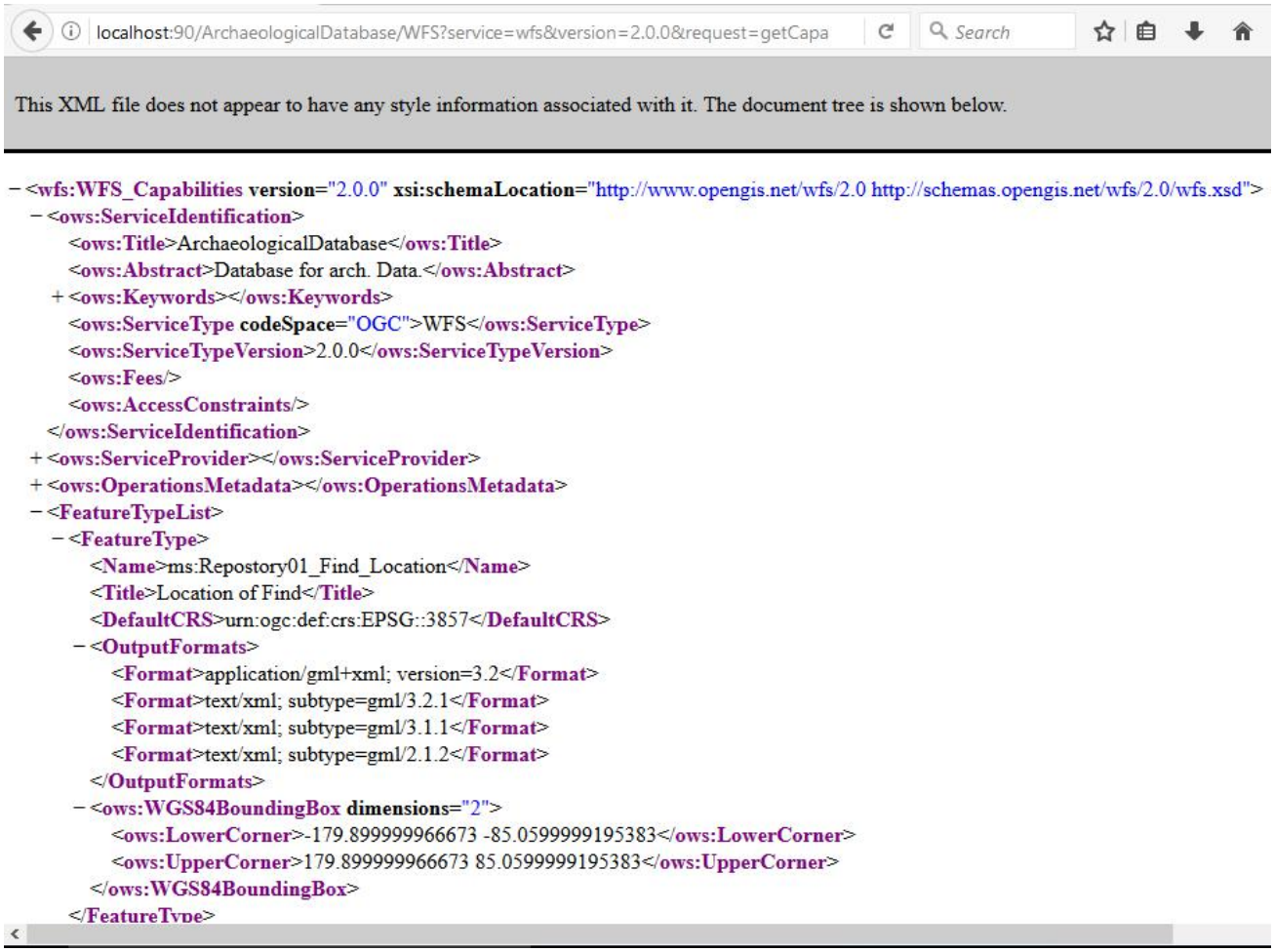


Figure 60

result of the GetCapabilities-request

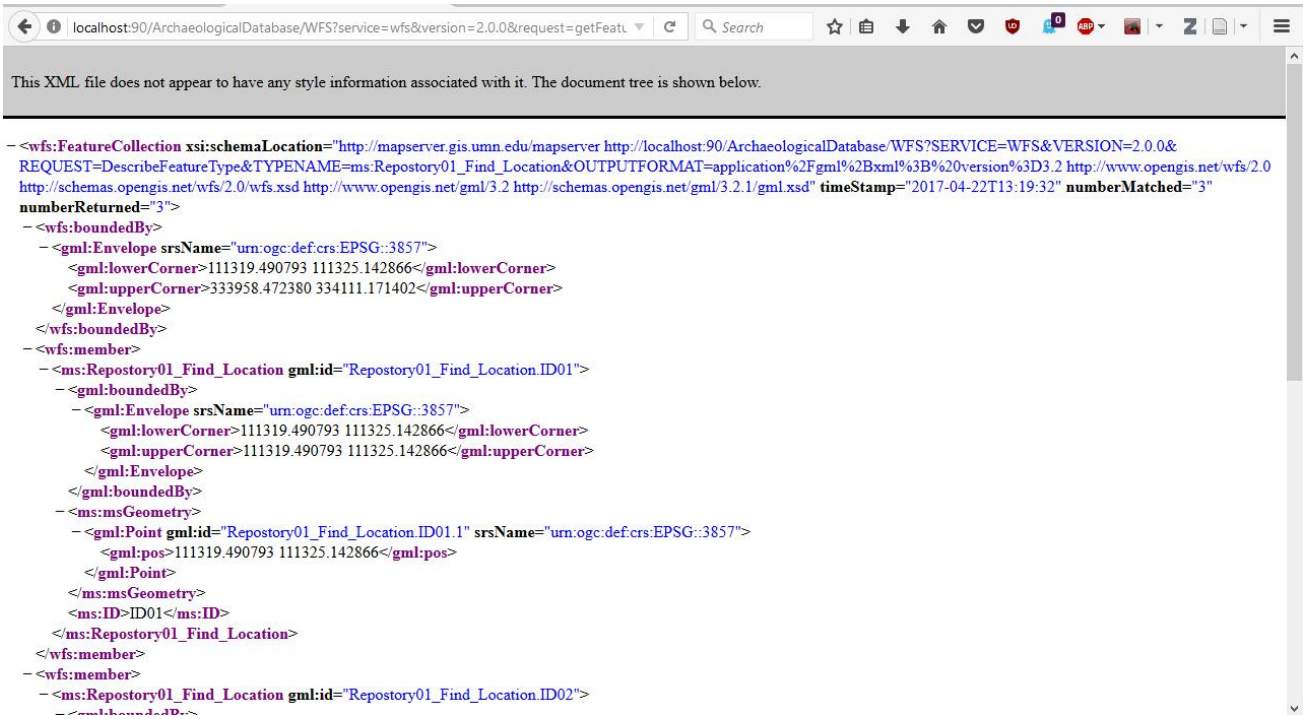


Figure 61

result of the GetFeature-request

A convenient characteristic of MapServer is the support for SpatiaLite by default, so that only the path to the database-file together with a corresponding SQL-statement must be declared. So as Figure 60 and Figure 61 demonstrate, the WebFeatureService works as expected, once all settings are accomplished.

Because it would be beyond the scope of this paper, no further attempts to refine the WebFeatureService is tackled at the moment, but it is to be stated that a script or module of the desktop-application could be generated taking care of automatically updating the map-documents

2.5.6. Results

2.5.6.1. Summary and Significance

As much as the desktop-application stands symbol for the process of generating data in the field, the components-bundle outlined in the past chapters is to represent the preliminary end of the data's journey within the whole software-architecture and in the course of this endeavour. Like outlined in the first chapter (see chapter 1. *Introduction*) the prime objective is to make excavations' (raw-) data available to interest-groups such as researchers and institutions in a comprehensibly defined form generally agreed on. While, as far as in this regard the nature of the generated data itself is concerned, the desktop-application already takes care of structure and semantics, the data-publishing-component is now in charge to communicate the information via likewise accepted channels. This is to say that the focus here does not rest any more with the data's actual ability to integrate with other excavations' akin data in terms of structure and meaning - since that is already accomplished -, but with defining the standards and technologies to apply for data-exchange, so that the recorded archaeological information can be analysed in the context of its domain with special emphasis on spatial aspects.The latter is to indicate the ambition to allow for loading the data into desktop- or web-GeographicInformationSystems, so as to bridge the gap between different excavation-projects in favour of an integrated analysis.

For that matter the undertaking needs to settle for standards defining data-transfer to consumers, which in this case is left to http/html, WebFeatureService and eXtensibleMarkupLanguage, because, on the one hand, a

website can be viewed on any device featuring a browser, while, on the other hand, WebFeatureService (using eXtensibleMarkupLanguage) basically represents structured text-documents accessible by virtually all computers and processible by common GeographicInformationSystems.

In order to fulfill those standards and protocols on the way to aforementioned overall objective, this endeavour is explicitly meant to derive advantage from already existing and approved technologies, that are in fact hardly subject to debate in the case of the website (utilizing html, javascript and cascading style sheets), but are chosen by fitness for server-side tasks employing an NGINX-server and PHP. By connecting the server with the open-source MapServer-software-bundle facilitating WebFeatureService all measures are finally assembled, in order to support a full work-flow for archaeological raw-data.

From the remote user's view this very work-flow starts with calling the website, that, first of all, intends to allow for browsing the different categories of information in an appearance suiting the data's character,[102] so that consequently the user gets an impression of the data he/she might want to obtain. Another important function of the website are its download-capabilities, that on the one hand equip the visitor with all the information necessary for setting up the exact same data-model together with the corresponding glossaries, but on the other hand offer the database-files themselves for download. Since the latter may be regarded a very open concept supposed to represent a paradigm of freely available archaeological data and, thus, a feature rather suitable for project internal data transfer, the access to certain downloads could easily be controlled by restricting the service to IP-addresses of local networks or by introducing a system of user-accounts.

Having gained an impression of the data-store through the website the user then can request and obtain the data via WebFeatureService.

[102] This simply means, that - in contrast to alphanumeric data - spatial data is displayed in maps and graphics, while the data-model is illustrated as a graph.

2.5.6.2. Conclusion

Irrespective of demonstrating a way of assembling a software-construct working as outlined in chapter *1.2.3. Conceptual Solution* this case-study adduced several findings, the most essential of which centers upon the database-system. Since for the data-store's backbone the file-based database-system SQLite is chosen over a database-server-application like PostgreSQL/PostGIS, various particularities must be considered: As the construct is not client to a database-server, but rather establishes database-connections itself by virtue of included libraries, it has to coordinate all database-related access-(rights-)management, which, however, regarding remote access is not much of a concern, because remote read-connections do not interfere harmfully with the single desktop-application's reading and writing. Through the chosen database-system, though, the engineered software-bundle is not only better capsuled and portable, but also bypasses a possible performance-bottleneck caused by an intermediary database-server. As a result, not only can database-files be made available for download, but the general pattern of excavation-data being usually organised in campaign-oriented groups is resembled by the structure of the applications. As a consequence, data is to be queried (on the website) and requested (via WebFeatureService) separately for each repository, which would be considered a downside, if this endeavour was expected to end in an extensive institution-level-datastore, while in fact it is only to be deployed at project scale.

As a means to promote a modular design and exchangeability of the major components, the only suitable touch points of desktop-application and server-related functionality are the resource-files, which means that the desktop-application does not intend to control the server, whereas the latter does its tasks using the resources it finds. The choice of the main server was geared by the simplicity - and, thus, portability - of NGINX, so that the server-component can be set up on any system simply by placing the server's files to the correct place of the software-architecture's directories-tree. As a supporting measure the server-side scripts are

constructed to look up dependencies relative to their own position rather than resorting to absolute paths.

In contrast to the server-side tasks having turned out pretty straightforward to implement a sophisticated client-side functionality required the most effort, so as to guarantee a satisfactory browsing experience. Like foreseen in the planning phase the challenge in implementing the website's capabilities is not so much rooted in developing all functions from scratch, but integrating existing JavaScript-frameworks, in order to exploit their functionality. In this process, however, an upload-mechanism is omitted, for it is not really required, whereby its absence eliminates some security-considerations completely. The ability to download referenced binary-files from the website, on the other hand, - as straightforward to implement as it is (see *GetFile.php*) - fulfills another essential task given the fact that archaeological records tend to be accompanied by photos and akin data. Since those binary files are stored outside the databases, file-size and database-connections are of no concern, so as to ease downloading.

Regarding the $W_{eb}F_{eature}S_{ervice}$ the available solutions turned out to provide functionality far beyond the requirements of this project, but nevertheless serve the purpose of having the $W_{eb}F_{eature}S_{ervice}$ installed as completely replaceable component. Employing a proxy mechanism redirecting requests to a local server hosting all $W_{eb}F_{eature}S_{ervice}$ related functions - as outlined in chapter *2.5.5. The WFS* - not only encapsulation is enforced, but the user requests can be furnished with additional parameters. Because acquiring the data can be achieved via SQL-statements, all sorts of table-joins can be made opening up the possibility to combine virtually all stored (or processed) data as attributes to geometries in stock or calculated. Due to Mapserver's paradigm of map-documents specifying the services, it allows for future automation. As a consequence the $W_{eb}F_{eature}S_{ervice}$ with its standardized capabilities is an ideal interface to spatial data - such as archaeological records - and may additionally serve well as source for *Research Infrastructures*, since the data-model available for download containing mappings to an ontology lets an infrastructure understand structure and meaning of the data-store.

3. RESULTS

3.1. VERIFYING RESULTS

In continuation of chapters *2.4.6. Results* as well as *2.5.6. Results* and in order to redirect the focus of discourse back to the initial subject of this paper - namely seeking to gain tremendous benefit from applying common digital technologies to archaeological data-management (in terms of excavations) -, the significance and results of this attempt are, at this point, illustrated best by mapping typical excavation-related tasks to the programs' capabilities. The latter are, thus, confronted with use-cases in so called use-case-diagrams each representing a campaign's main stage ordered chronologically in the manner of a story-board.

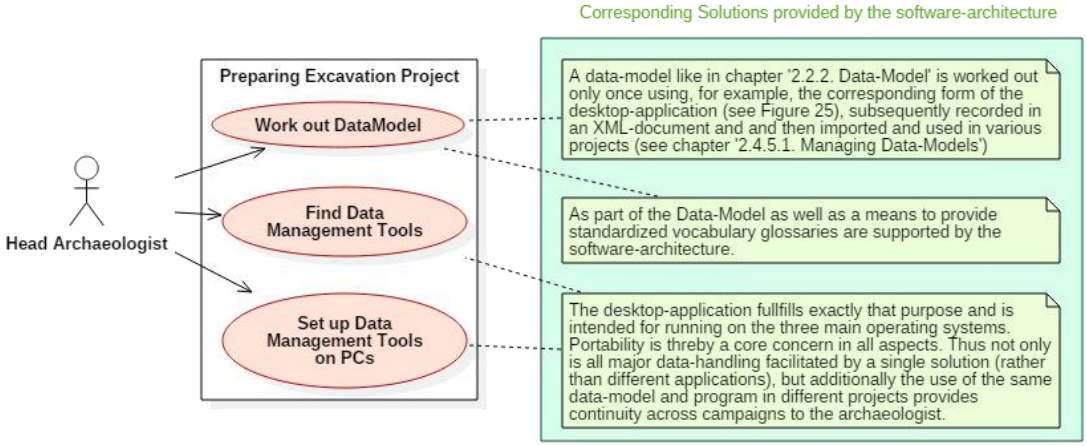


Figure 62

Use-Case highlighting tasks of the pre-project phase

Figure 62 displays tasks involved in preparing an excavation-project requiring the set-up of data-handling-programs, which is exactly this software's intended field of application. Thus, instead of assembling ad-hoc a bunch of rather misused programs with each project starting, this very software specialized in archaeological data-management is already prepared to be applied again and again in different campaign. Also the data-model, once created, is intended for repeated use.

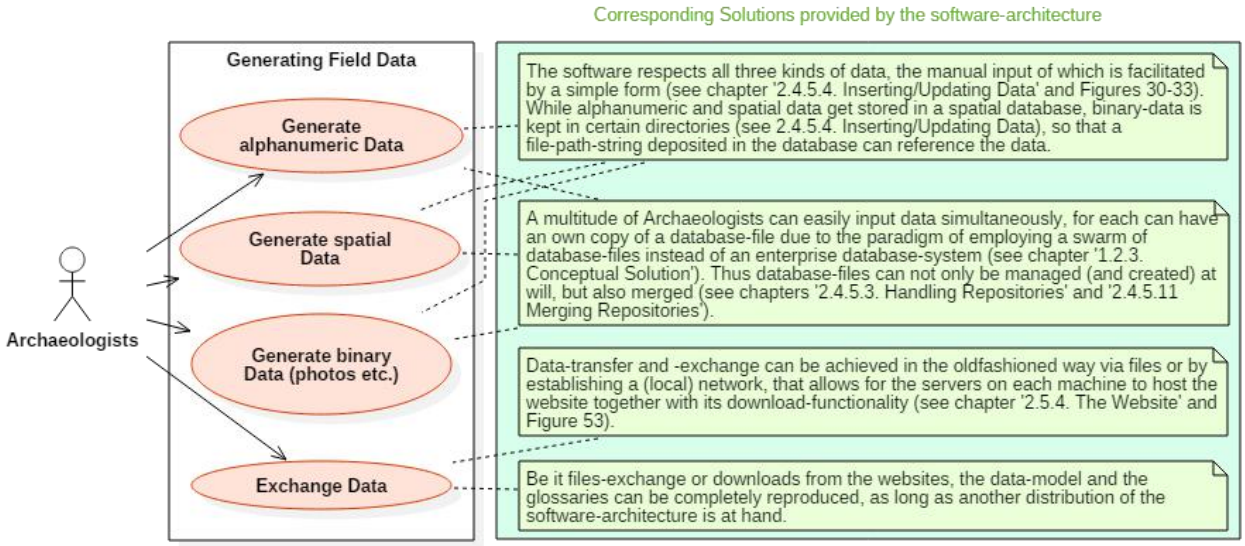


Figure 63

Use-Case highlighting tasks of data-recording

During the excavation-process the most important aspects are unhindered simultaneous access to data-entry as well as sharing and exchanging the data, which is all met by the program, as Figure 63 demonstrates. The program's focus on portability makes its transfer to and application on different machines easy, which is also supported by external binary files being stored close to the databases and addressed via relative paths. An apparently still common and classic approach to data-handling encompassing data-storage in disperse files - such as Microsoft Word/Excel-files for alphanumeric data and GIS-program-specific files for spatial data - is explicitly sought to be avoided.

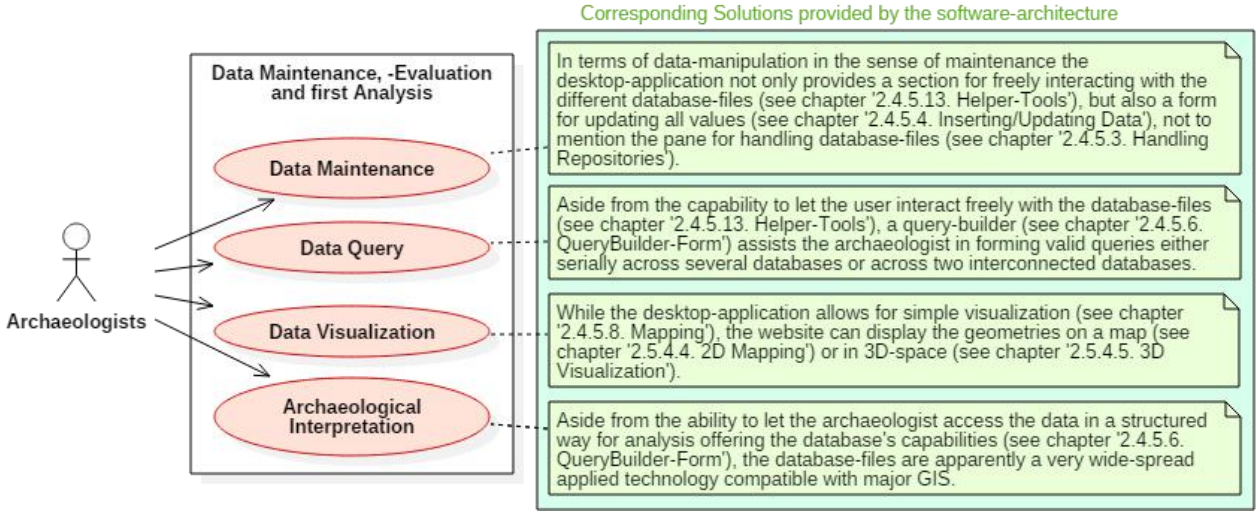


Figure 64

Use-Case highlighting tasks of database-interaction after the actual excavation

The use-cases of Figure 64 refer to activities beyond - respectively after - the sheer data-collecting and target the issues of archaeological analysis. The intention of the software-architecture here is to provide (by default) a maximum of functionality such as the full functional range of a spatial database-system as well as visualisation tools in an immediately accessible, but nevertheless generic way, so as to support a multitude of data-models. Due to the programs ability for downloading, exporting and copying database-files (see chapters *2.4.5.3. Handling Repositories* and *2.5.4. The Website*) together with the database-system's apparent popularity leading towards file-format-compatibility with common $G_{eographic}I_{nformation}S_{ystems}$ archaeologists get the chance to work with the data simultaneously employing also, for instance, desktop-$G_{eographic}I_{nformation}S_{ystems}$. As it hence turned out, the software is obliqued to provide tools to overcome data-segregation (across files) and reintegrate database-files (see chapters *2.4.5.3. Handling Repositories*, *2.4.5.11. Merging Repositories* and *2.4.5.12. External Indices*).

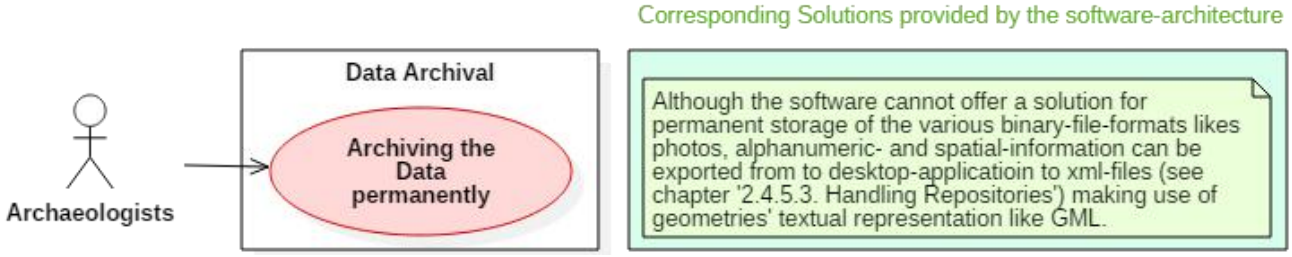


Figure 65

Use-Case highlighting task of archiving archaeological data

In order to allow for archiving the data persistently (in digital terms) the software is capable of generating simple text-files containing the alphanumeric and spatial data in a non-proprietary format.

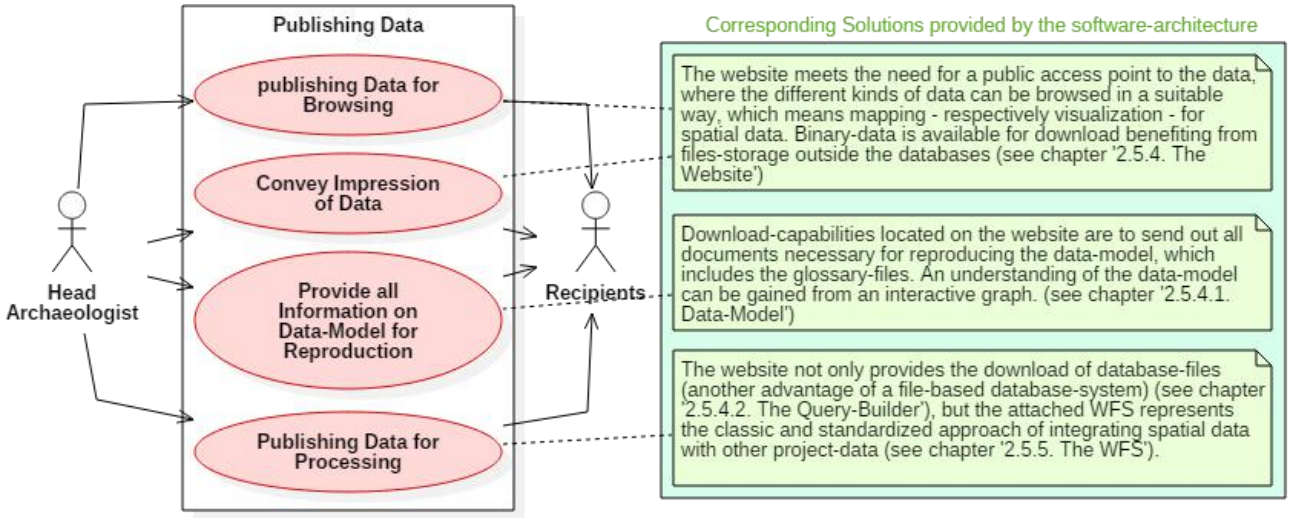


Figure 66

Use-Case highlighting tasks of publishing archaeological data

In order to publish the data on a local network or the internet, a server is required, which in this case hosts a website and redirects $W_{eb}F_{eature}S_{ervice}$-requests to another (local) server specialized in handling $W_{eb}F_{eature}S_{ervice}$-tasks. Since the main purpose of this work is to develop a software-bundle facilitating archaeological (raw-) data's life-cycle, the $W_{eb}F_{eature}S_{ervice}$ fulfills the final requirement. All those published resources combined with the data.model referencing an ontology offer themselves as data-provider in a *Research Infrastructure*. The necessary metadata can be gained from a repository-query (like outlined in chapters *2.5.4.2. The Query-Builder* and *2.5.4.3. Query-Results*).

Conclusively, comparing the archaeological use-cases with the implemented functionality of the software architecture ascertained the fact, that all major aspects of archaeological data-management are actually addressed by the software and indeed supplied with a solution thus verifying this case-study having reached its final goal. Since chapter *1.2.3. Conceptual Solution* presents a theoretic solution, the outcome of this paper may also be rated in terms of having actually implemented the required functionality, which is indeed true.

3.2. Exemplifying a practical Use-Case

Given a typical (but hypothetical) excavation-scenario comprising the unearthing of a pit (containing a vessel) this chapter intends to illustrate the functioning of the software in the various stages of the archaeological process. In doing so not all tables of the data-model need to be filled with values, but only a subset (see chapter *3.2.3. Data-Maintenance and -Handling*) sufficing to demonstrate the software's qualities.

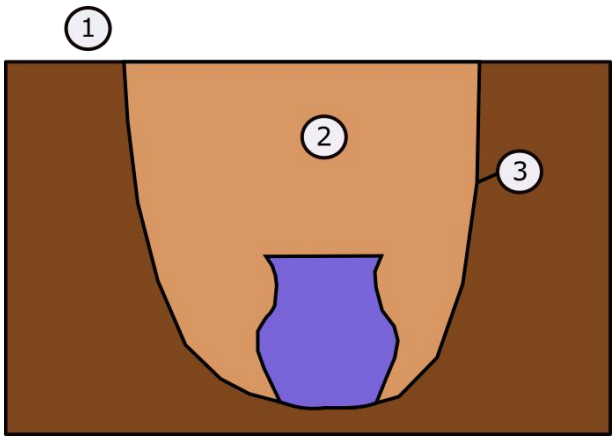


Figure 67

test-scenario

In regard of above shown use-cases (see *3.1. Verifying Results with Use-Cases*) this example omits most of the actions relating to preparing the excavation, since due to software and data-model already existing it only remains to import the $_{e}X_{tensible}M_{arkup}L_{anguage}$-document holding the data-model (see chapter *2.4.5.1. Managing Data-Models*) into the desktop-application and fill the glossaries with entries in the manner illustrated in chapter *2.4.5.5. Inserting/Updating Glossaries*. (In a real scenario this paper assumes data-model and glossaries to be given by an agency.)

The sub-chapters of this section correspond with the remaining use-case-groups of the preceding chapter.

3.2.1. Employed Glossaries

Where the *Guidelines for Archaeological Measures in Austria* do not provide the terminology to fill the fields with (multiple-choice-fields) and a systematic nomenclature is clearly desired, vocabularies are sought to be employed, so as to purvey documented and recognised terminologies.

This is the case concerning characterisation of sediment by its texture: In accordance with a procedure called “Fingerprobe” a system of soil types[103] was cast into a standard[104], that provides a frame of reference for the data model. Since the id-number of the particular “Katastralgemeinde”[105] the excavation takes place at is always part of the project-id (“Maßnahmennummer”), a table supplying the corresponding name and number is incorporated into the data-model. A similar measure is applied to a table holding the names for anatomic skeletal parts of the human body, corresponding terminologies of which can easily be found, as it is common knowledge.[106] If possible, wooden structures are supposed to be identified in matters of their type, respectively tree species. Although probably overburdening the archaeologist, the terminology is provided in the data model.[107] In archaeology colour is usually expressed in the codes of the Munsell Soil Color Charts[108], so that they are given a table of their own. While it appears rather hard to find acknowledged german archaeological thesauri, the cultural heritage sector of the United Kingdom offers a rich source in this regard.[109] Thus the created data model makes use of that resource by applying the provided vocabulary on structural components[110] allowing remains of buildings uncovered by excavation to be properly classified. The same approach is adopted to rock types[111] representing building material.

The discussion on the nomenclatures utilised in the data model reveals the patchy nature of the archaeological domain regarding its terminologies. The latter are, in fact, obviously frequently either borrowed from other domains or stand for common knowledge. Extensive collections of terms - as they are established in the United Kingdom’s cultural heritage - represent a rather rare condition. Hence, it turns out virtually impossible to obtain all the vocabularies needed from a single officially approved source.

---

[103] Caution: In German there is a difference between “Bodenart” and “Bodentyp”.

[104] DIN 19682-2: 2014 07: acquirable at: https://shop.austrian-standards.at/action/de/public/ details/524449/DIN_19682-2_2014_07 <last visited 13.11.2015>. Also see: https://de.wikipedia.org/wiki/Fingerprobe_%28Boden%29 <last visited 13.11.2015> and https://www.lko.at/?+Fingerprobe-ermittelt-Bodenart+&id=2500,2157812,,,,bW9kZT1uZXh0JnBhZ2luZz15ZXNfXzMwJmN0PTMzJmJhY2s9MQ <last visited 13.11.2015>. For a discussion of a similar model in archaeological terms see Möller 2001.

[105] published by the “Bundesamt für Eich- und Vermessungswesen” (http://www.bev.gv.at/pls/portal/docs/PAGE/BEV_PORTAL_CONTENT_ALLGEMEIN/0200_PRODUKTE/UNENTGELTLICHE_PRODUKTE_DES_BEV/KG-Verzeichnis.zip <last visited 20.10.2015>) and free to use under following conditions: http://www.bev.gv.at/pls/portal/docs/PAGE/BEV_PORTAL_CONTENT_ALLGEMEIN/0200_PRODUKTE/BESTELLFORMULARE/STANDARDENTGELTE_UND_NUTZUNGSBEDINGUNGEN_2015.PDF<last visited 20.10.2015>.

[106] Here Platzer – Leonhardt 2005 is used.

[107] Source: http://www.baumkunde.de/baumlisten/baumliste_az_scientific.php <last visited: 14.11.2015>.

[108] Here: Munsell, *Soil Color Charts* (Michigan 2009).

[109] See: http://thesaurus.historicengland.org.uk/frequentuser.htm <last visited: 14.11.2015>.

[110] Source of thesaurus: http://thesaurus.historicengland.org.uk/class_list.asp?thes_no=546&class_no=180054&class_name=STRUCTURAL%20COMPONENT <last visited: 14.11.2015>.

[111] Source of thesaurus: http://thesaurus.historicengland.org.uk/thesaurus.asp?thes_no=129&thes_name=FISH%20Building%20Materials%20Thesaurus <last visited: 14.11.2015>.

3.2.2. Inserting Field Data

After the data-model is set active (like in Figure 24) in the desktop-application, one or more repositories are created (like in Figure 28), which means that aforesaid glossaries transfer their values into the database-file. In case there are several archaeologists inserting data simultaneously or the huge number of glossary-values slows down the repository-creation-process, the first repository may serve as template for creating further repositories via copying, exporting or download (see chapters *2.4.5.3. Handling Repositories* and *2.5.4. The Website*), so as to provide co-workers with a starting-point for their data-entry. For the data-entry-process itself utilising the data-entry-form outlined in chapter *2.4.5.4. Inserting/Updating Data* the archaeologist works his/her way up the list of tables (as shown in Figure 26), so as to fill in values in the correct order and have, thus, foreign-key-values at hand as dropdown-menu when needed. In this example it starts with the projects-table and goes as far as the tables USSediment, USInterface and Find.

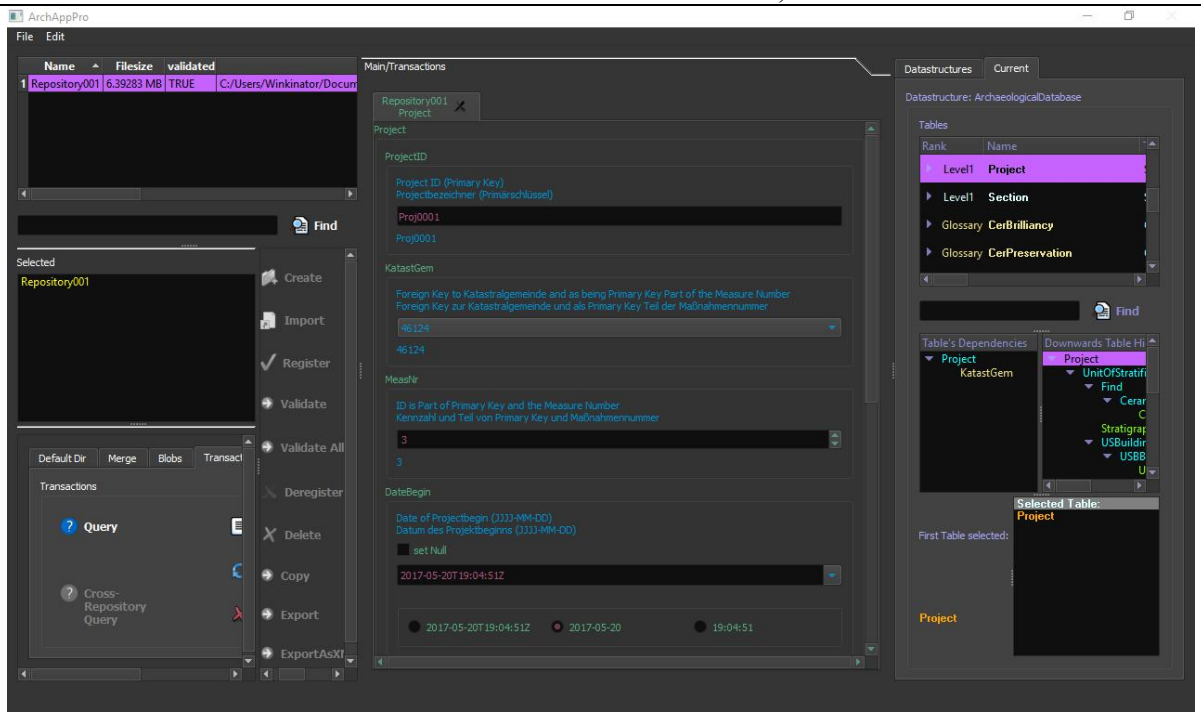


Figure 68

inserting data into the Projects-table using the pane for data-entry (see chapter 2.4.5.4. Inserting/Updating Data)

In order to manage binary-data like images, text-fields allow to select a file-path, so that the file is copied to the repositories folder and the new path stored in the text-field:

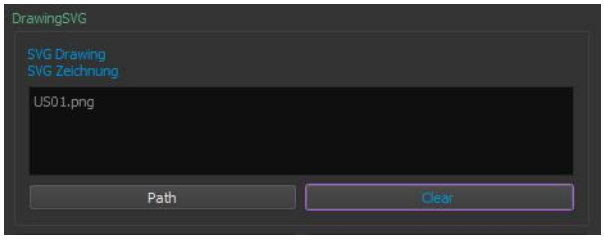


Figure 69

textfield with path

Geographic data is inserted the following way resulting in a $W_{ell}K_{nown}T_{ext}$-string:

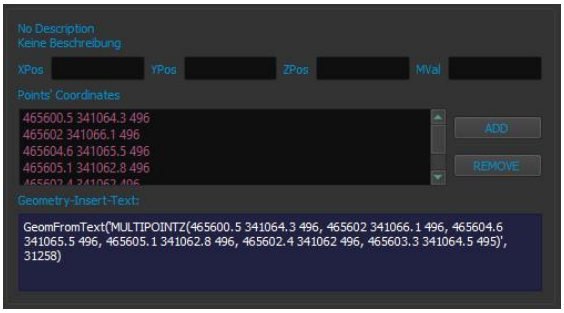


Figure 70

geometry-input-field

The result of these efforts is a repository-database-file accompanied by a folder of binary-data, which both - in case data-entry was done simultaneously by several archaeologists with several repositories - can be merged with other repositories of the same data-model (see *chapter 2.4.5.11. Merging Repositories*) or shared with others by using the export- or copy-functions of the desktop-application (see chapter *2.4.5.3. Handling Repositories*).

3.2.3. Data-Maintenance and -Handling

Now that data is stored, one might want to change, delete or update it, the latter of which being exemplified by Figure 71 employing the form introduced in chapter *2.4.5.4. Inserting/Updating Data*. In this case the location of the finds' storage is changed to 'Stadtmuseum' in the projects-table.

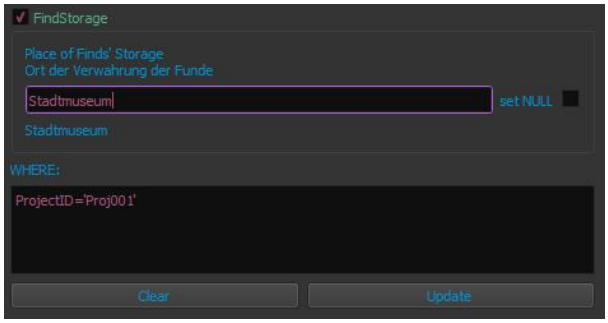


Figure 71

Update-panel with WHERE-condition and Check-Box for specifying the values to be updated

More important, however, sure is the capability to query the data in the user-friendly way outlined in the chapters *2.4.5.6. QueryBuilder-Form* and *2.4.5.7. Query Results*. The following figures exemplify the query-results of tables endowed with values:

Figure 72

results of querying table Project

Figure 73

results of querying table Person

Figure 74

results of querying table ObjectGroup

Figure 75

results of querying table Object

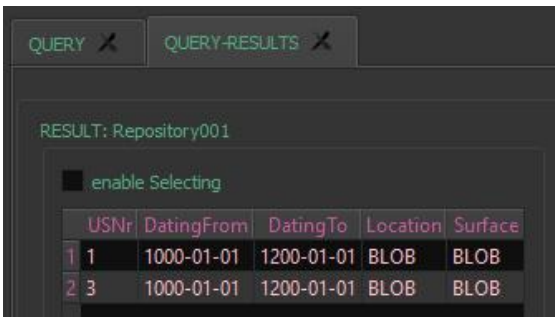


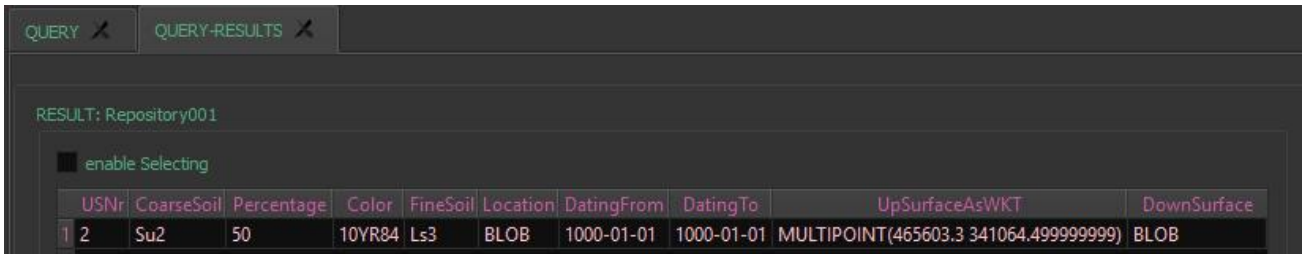
Figure 76

results of querying table USInterface

	USNr	USTerm	ProjectID	Operator	RecordingDate	Interpretation	ObjectNr	LinkToModel	DrawingSVG
1	1	Surface	Proj001	1	2017-05-27	Ground	1	NULL	US01.png
2	2	Pitch-Filling	Proj001	1	2017-05-28	Sediment-Filling of the Pitch	1	NULL	US02.png
3	3	Pitch	Proj001	1	2017-05-29	Pitch	1	NULL	US03.png

Figure 77

results of querying table UnitOfStratifiaction



	USNr	CoarseSoil	Percentage	Color	FineSoil	Location	DatingFrom	DatingTo	UpSurfaceAsWKT	DownSurface
1	2	Su2	50	10YR84	Ls3	BLOB	1000-01-01	1000-01-01	MULTIPOINT(465603.3 341064.499999999)	BLOB

Figure 78

results of querying table USSediment

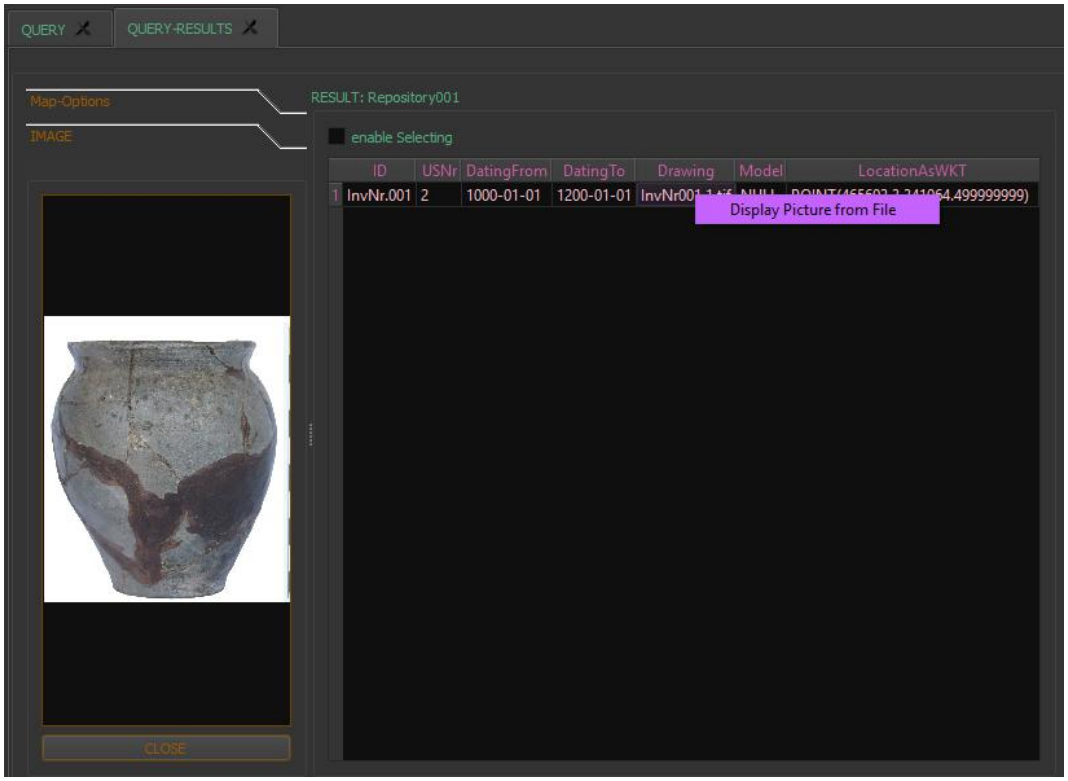


Figure 79

results of querying table USFind with display of a referenced photo

As illustrated by Figure 79 the query-results are also the starting point for visualisation of referenced images or geometry, just like Figure 80 and Figure 81 exemplify the local mapping process.

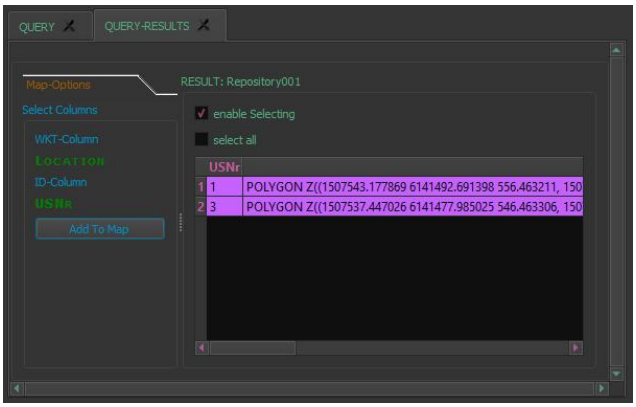


Figure 80

query-results of table USInterface containing geometry

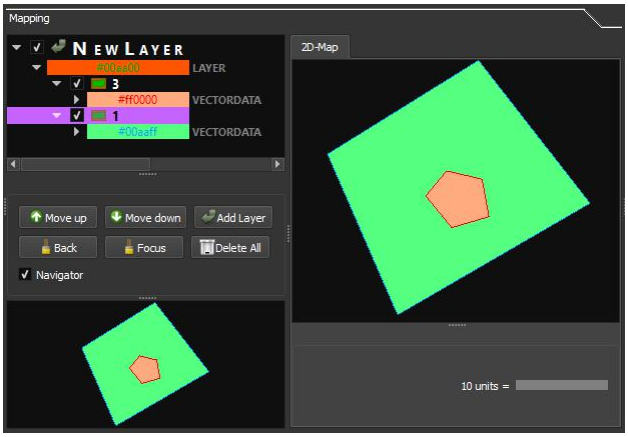


Figure 81

mapping Location from Table USInterface

As is well known, SpatiaLite-files can be accessed by QGIS hence opening up the option of sophisticated geospatial analysis to archaeologists:

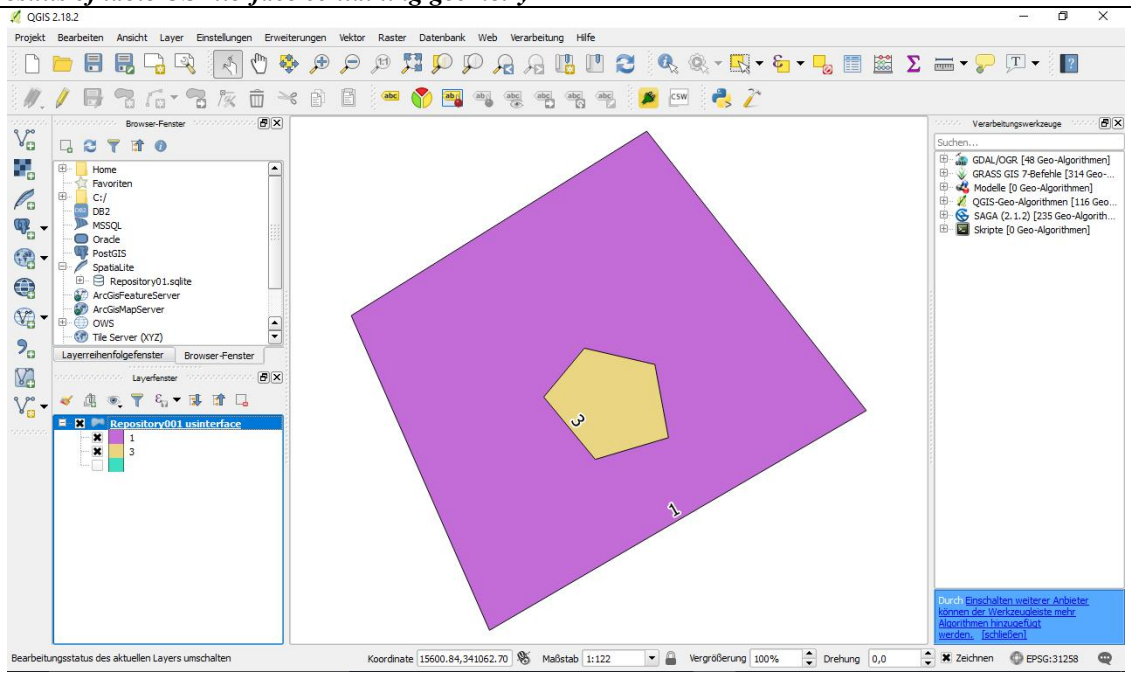


Figure 82

repository directly loaded as SpatiaLite-Database into QGIS displaying Locations from fable USInterface

3.2.4. Data Archival

Like outlined earlier long-term data-storage is definitely an issue in archaeology leading towards the mechanism introduced in chapter *2.4.5.3. Handling Repositories*, that exports alphanumeric data and geometry as $_{e}X_{tensible}M_{arkup}L_{anguage}$-documents together with copies of the binary-files (referenced and stored by the repositories). The generated file-structure containing the tables' content as XML- and binary-files is represented by Figure 83 and Figure 84.

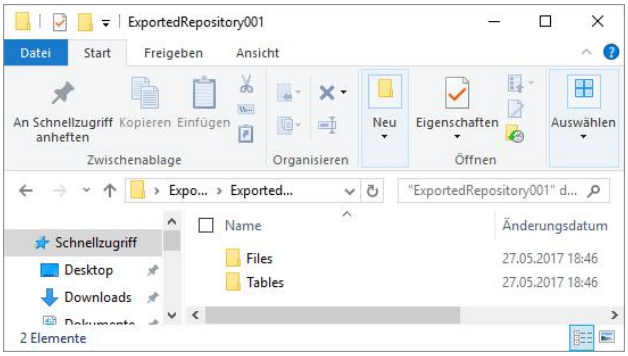


Figure 83

root-directory of the destination-folder

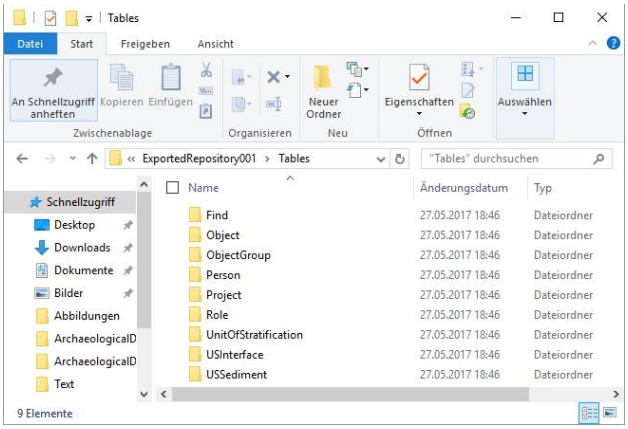


Figure 84

content of the Tables-directory

Following is the content of the USSediment eXtensibleMarkupLanguage-File:

```
<?xml version="1.0" encoding="UTF-8"?>
<Table Name="USSediment"
xmlns="http://www.thomaswinklehner.at/Glossary"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:schemaLocation="http://www.thomaswinklehner.
at/Glossary GlossarySchema.xsd">
 <Row>
  <PrimaryKeyColumn Name="USNr">2
  </PrimaryKeyColumn>
   <Column Name="CoarseSoil">Su2</Column>
   <Column Name="CoarseSoilPercentage">50
   </Column>
   <Column Name="FineSoil">Ls3</Column>
   <Column Name="Color">10YR84</Column>
   <Column Name="PrelimDatingFrom">1000-01-01
   </Column>
   <Column Name="PrelimDatingTo">1000-01-01
   </Column>
   <Column Name="Location">POLYGON((465600.5
341064.2999999998,465601.9999999999
341066.0999999986,465604.5999999999
341065.5,465605.1
341062.7999999998,465602.3999999999
341061.999999999,465600.5
341064.2999999998))</Column>
   <Column Name="UpSurface">MULTIPOINT(465603.3
341064.499999999)</Column>
   <Column
Name="DownSurface">MULTIPOINT(465600.5
341064.2999999998,465601.9999999999
341066.0999999986,465604.5999999999
341065.5,465605.1
341062.7999999998,465602.3999999999
341061.999999999,465603.3
341064.499999999)</Column>
 </Row>
</Table>
```

3.2.5. Publishing Data

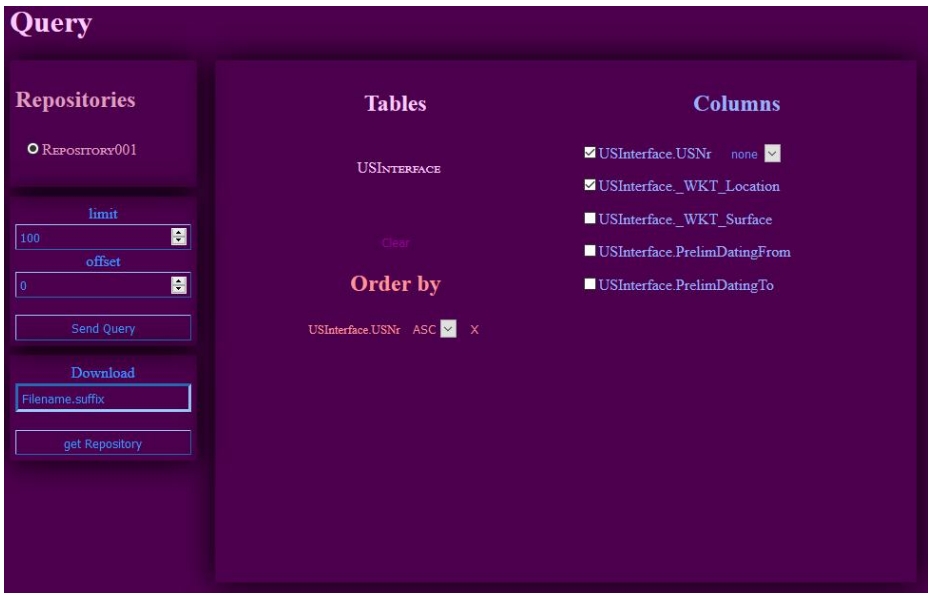


Figure 85

query for retrieving Project-data

Retrieving data from the website starts with launching queries (Figure 85), as shown in chapters *2.5.4.2. The Query-Builder* and *2.5.4.3. Query-Results* (thus leaving aside data-model-related information for the moment), which in this case leads towards the following output for the table USInterface:

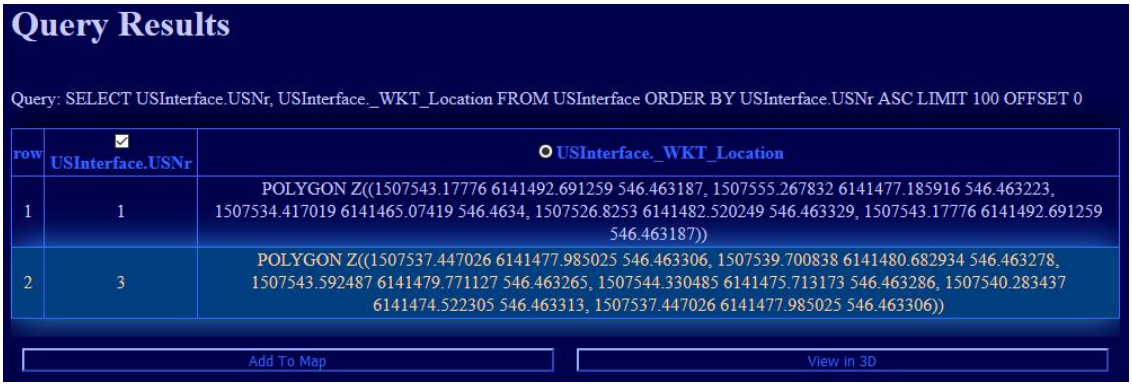


Figure 86

query-results for USInterface

From here the geometry can be selected for display on the map (as outlined in chapter *2.5.4.4. 2D Mapping*), where the spatial data is organised in layers, whose colours can be modified. Together with the automatic labels and the addable WMS/WFS-layers a user gets an impression of the spatial extent of the features and can set them in context with other mapped data like in Figure 88.

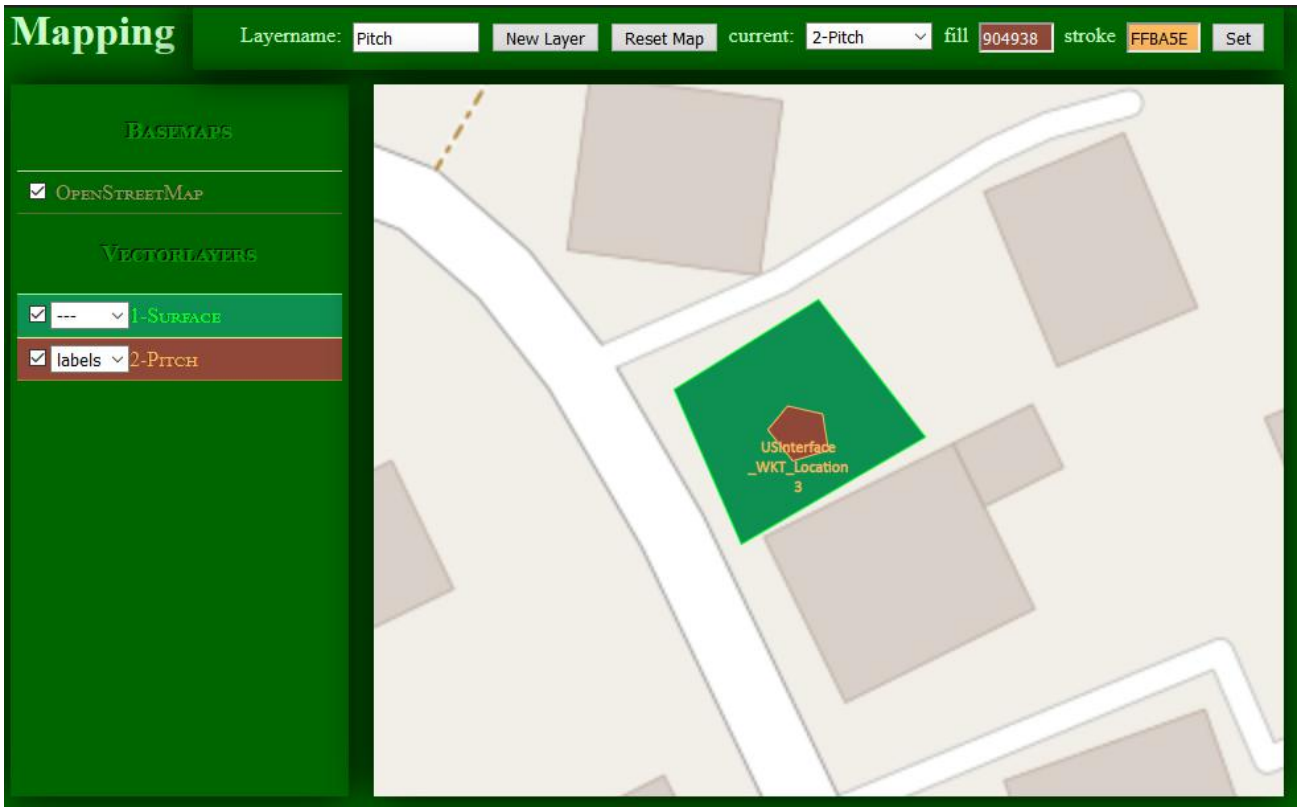


Figure 87

mapping USInterfaces with OpenStreetMap-basemap

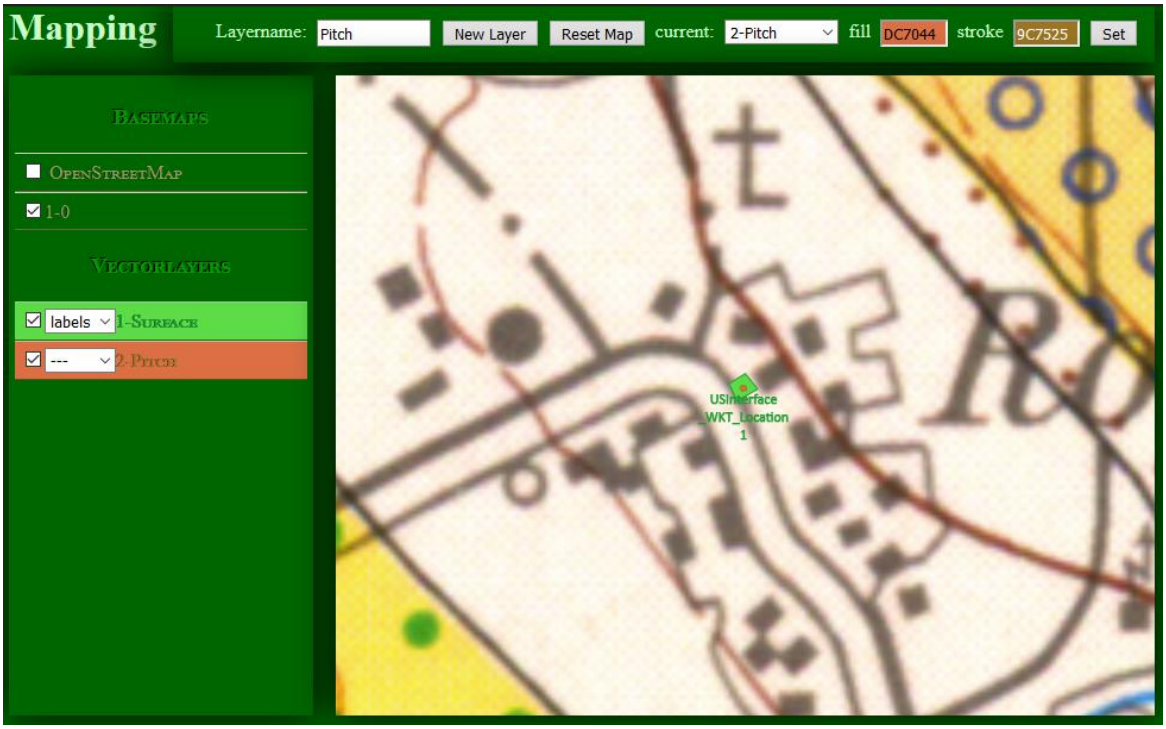


Figure 88

mapping USInterfaces with geological basemap included via WMS

As a means to investigate the three-dimensional relations of spatial features, the 3D Mapping pane allows for viewing geometry in 3D-space, which is a central issue in archaeology.

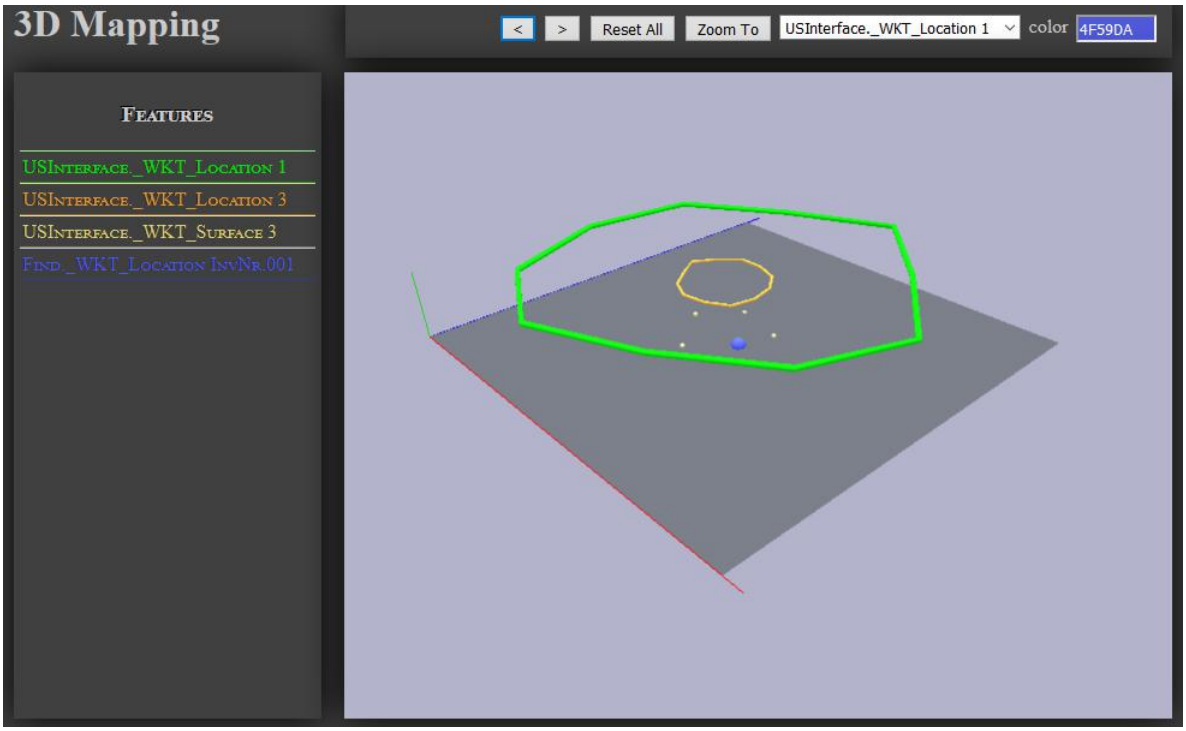


Figure 89

features of USInterface and Find displayed on 3D-mapping-pane

Although the 3D-Mapping-pane only provides a very basic functionality and is sure to be elaborated further for practical usage (adding, for instance, tessellated surfaces), the technical possibilities applicable to the benefit of archaeology are already apparent.

The next capability to highlight is the download of binary-files referenced by path in a database-field, which is accomplished by selecting the repository in the website's query-area and specifying the file-name:

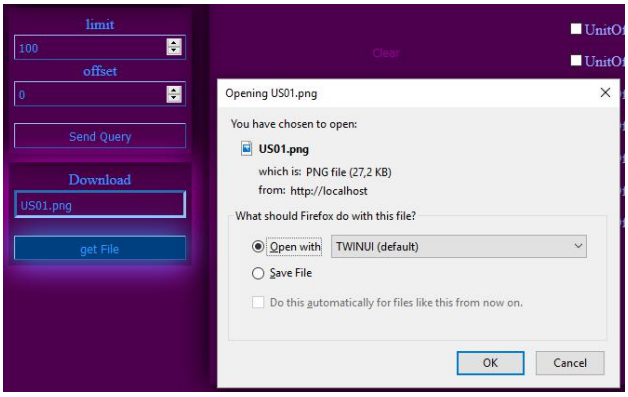


Figure 90

download of binary-file

With this mechanism finally even binary data is delivered to the user thus completing the range of the website's functionality in terms of publishing the stored data. As an additional feature, however, all data-model related documents can be viewed in the browser by the user, as exemplified by Figure 91 displaying the data-model in the browser.

Figure 91

data-model displayed in browser

As the software's final feature it remains to present the WebFeatureService in action utilising QGIS's capability for loading data via WFS. As Figure 92 and Figure 93 illustrate, there is absolutely no problem loading geometry and attributes:

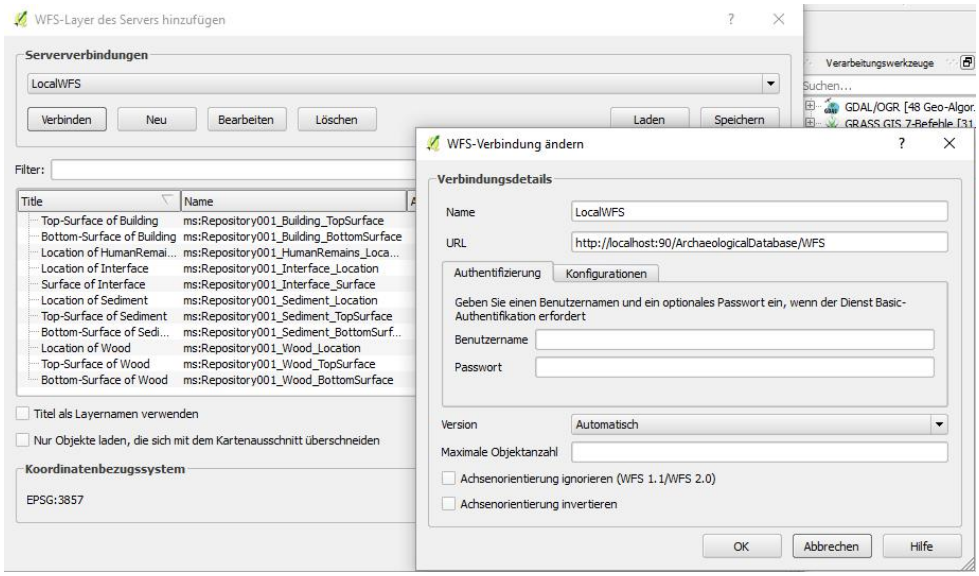


Figure 92

settings for loading WFS in QGIS

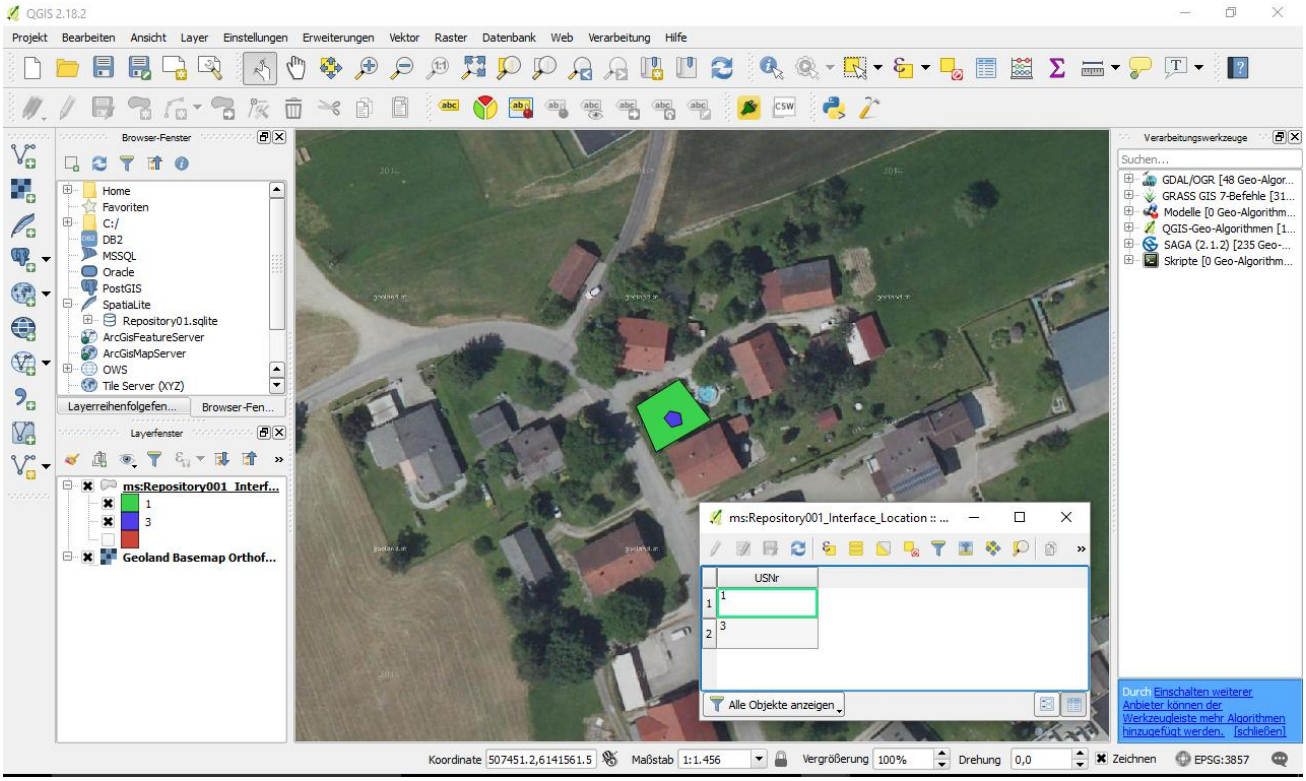


Figure 93

data loaded into QGIS via WFS with basemap

In a nutshell, this chapter demonstrated that, indeed, the software works as intended and provides several useful features to the archaeologist out of the box and ready for usage. Although there is still much place for enhancements - especially with the website representing more of an exemplary pattern for implementation -, important insights were gained from the implementation, that can be applied in an industry-standard version hereafter.

4. DISCUSSION

After having reviewed the results of this endeavour and gazed at the consequent implications at various occasions (see chapters *2.4.6. Results*, *2.5.6. Results* and *3. Results*), it remains to summarize the significance of this work in the light of the overall subject. Like outlined in the introductory chapter (*1. Introduction*) this paper's goal is to develop and provide a method to overcome diversification of archaeological raw-data originating from different excavations in Austria and stimulate data-integration, so as to allow for collective data-analysis - especially in spatial terms.

The first thing to notice about the taken approach is the act of transferring the guidelines for excavations into a data-model representing the key to collective data-integration. By this means each excavation-campaign no longer has to worry about coming up with a strategy for structuring records, but can simply resort to the data-model with its glossaries leading towards standardized records, that can effortlessly be merged. For integration on a more abstract level the data-model can additionally be enriched with mappings to an ontology.

Since the current guidelines - respectively the data-model - is mandatory for excavations in Austria and requirements to data-management are more or less similar in different campaigns, a program collectively applied at the various excavations can not only provide the desired functionality of data-input, -sharing,- analysis and -publishing thence freeing archaeologists from assembling those tools ever anew , but - above all - guarantee the validity of the data in terms of the data-model. The developed software is intended to take that very role, so as to let archaeologists conduct excavation in the usual manner, but to draw way more use from the recorded data. The benefit, therein, not only is to be the expense-savings resulting from having a specialised software on hand ready for data-storage, but the fact that the data is fed into an infrastructure automatically packaging the records into database-files and publishing them via acknowledged protocols such as HTTP and WFS. Through that not only do other archaeologists get access to digital raw-data - a circumstance apparently hardly prevailing -, but also the option arises to participate in *Research Infrastructures*. In these settings the archaeologist is additionally enabled to obtain other projects' data, in order to immediately set the own campaign's findings in context with the state of research.

Two things, however, are to be stressed: The implementation of the website and the WFS - although fully functional - may be solved on other servers in different ways, so that the strategy taken in this work is to be understood merely as a suggestion. Moreover, the software presented in this paper is not intended to serve as centralized data-store in the scale of an institutional-level framework, but has its field of application mostly in the project-scale.

BIBLIOGRAPHY

Anyuru 2012 — A. Anyuru, *Professional WebGL Programming: Developing 3D Graphics for the Web*, Wrox Programmer to Programmer (Chichester: Wiley, 2012).

Apollonio and others 2010 — F. Apollonio and others, 'An Integrated 3D Geodatabase for Palladio's Work', *International Journal of Architectural Computing*, 8.2 (2010).

Aspöck – Geser 2014 — E. Aspöck, G. Geser, and others, 'What Is an Archaeological Research Infrastructure and why do we need it? Aims and Challenges of ARIADNE', in *Proceedings of the 18th International Conference on Cultural Heritage and New Technologies 2013 (CHNT 18, 2013)*, 2014, XVIII <http://www.chnt.at/wp-content/uploads/Aspoeck_Geser_2014.pdf>.

ARIADNE — http://www.ariadne-infrastructure.eu/ <last visited: 16.10.2015>.

AVROSS — F. Barjak and others, *Accelerating Transition to Virtual Research Organisation in Social Science (AVROSS)*, Brussels: Information Society and Media Directorate General (Brussels: Commission of the European Communities, 2007) <http://web.fhnw.ch/plattformen/avross/papers-and-prensentations/final-report/Final%20report,%20full%20draft,%20v2-2.pdf>.

Barker 1999 — R. Barker, *CASE Method: Entity Relationship Modelling*, Computer Aided Systems Engineering, Reprint (Harlow: Addison-Wesley-Longman, 1999).

BDA — http://www.bundesdenkmalamt.at/documents/621701608.pdf <last visited: 5.10.2015>.

Billen and others 2013 — N. Billen and others, 'Development of a 4D-Webgis for Archeological Research', in *Workshop on Integrating 4D, GIS and Cultural Heritage, 16th AGILE 2013* (Leuven, 2013) <http://mayaarch3d.org/wordpress/wp-content/uploads/2013/07/4D-webgis_for_archaeological.pdf>.

Berners-Lee and others 2001 — T. Berners-Lee and others, 'The Semantic Web', *Scientific American*, 284.5 (2001) <http://www-sop.inria.fr/acacia/cours/essi2006/Scientific%20American_%20Feature%20Article_%20The%20Semantic%20Web_%20May%202001.pdf>.

Blanchette – Summerfield 2006 — J. Blanchette and M. Summerfield, *C++ GUI Programming with Qt 4* (Upper Saddle River, NJ: Pearson Hall in association with Trolltech Press, 2006).

Brinkhoff 2013 — T. Brinkhoff, *Geodatenbanksysteme in Theorie und Praxis. Einführung in objektrelationale Geodatenbanken unter besonderer Berücksichtigung von Oracle Spatial*, 3rd edn (Berlin: Wichmann, 2013).

Brovelli – Magni 2003 — M. A. Brovelli and D. Magni, 'An Archaeological Web GIS Application Based on Mapserver and PostGIS', *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34.5/W12 (2003) <http://www.isprs.org/proceedings/XXXIV/5-W12/proceedings/19.pdf>.

Cefalo and others 2011 — R. Cefalo and others, 'Integrated Topographic, GNSS, Remote Sensing and GIS/WebGIS Techniques Applied to the Study of Aquileia River Port Structures', in *The New Technologies for Aquileia* (Aquileia, 2011) <http://ceur-ws.org/Vol-806/paper6.pdf>.

Costa and others 2014 — L. Costa and others, 'Integration of Archaeological Datasets through the Gradual Refinement of Models', in *Proceedings of the 42nd Annual Conference on Computer Applications and Quantitative Methods in Archaeology* (presented at the CAA 2014, Paris, 2014) <http://www.researchgate.net/profile/Cesar_Gonzalez-Perez/publication/261709682_Integration_of_Archaeological_Datasets_Through_the_Gradual_Refinement_of_Models/links/55b0baa608ae11d31039ce69.pdf>.

$CRM_{archaeo}$ — http://www.ics.forth.gr/index_main.php?l=e&searchtype=st_p&c=711 <last visited: 16.10.2015>.

DAHRIA-EU 01 — https://dariah.eu/ <last visited: 04.10.2015>.

DAHRIA-EU 02 — DARIAH-EU Coordination Office, *Introducing DARIAH-EU*, 2013 <https://dariah.eu/fileadmin/Documents/Library/DARIAH-EU_Information_brochure_autumn_2013.pdf>.

Davis 2004 — S. R. Davis, *C++ for Dummies*, 5. ed (Indianapolis, Ind: Wiley, 2004).

Deitel – Deitel 2012 — P. J. Deitel and H. M. Deitel, *C++: How to Program*, How to Program Series, 8th ed (Boston:

Prentice Hall, 2012).

Denkmalschutzgesetz — https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10009184 <last visited: 13.05.2017>.

De Luca and others 2011 — L. De Luca and others, 'A Semantic-Based Platform for the Digital Analysis of Architectural Heritage', *Computers & Graphics*, 35.2 (2011).

DFG 2013 — *Vorschläge zur Sicherung guter wissenschaftlicher Praxis: Denkschrift ; Empfehlungen der Kommission 'Selbstkontrolle in der Wissenschaft' = Proposals for Safeguarding Good Scientific Practice: Memorandum ; Recommendations of the Commission on Professional Self Regulation in Science*, ed. by Deutsche Forschungsgemeinschaft, Erg. Aufl. (Weinheim: Wiley-VCH, 2013) <http://www.dfg.de/download/pdf/dfg_im_profil/reden_stellungnahmen/download/empfehlung_wiss_praxis_1310.pdf>.

Doerr and others 2004 — M. Doerr and others, 'Integration of Complementary Archaeological Sources', in *Beyond the Artefact – Digital Interpretation of the Past. Proceedings of the 32th International Conference on Computer Applications and Quantitative Methods in Archaeology* (presented at the 32th International Conference on Computer Applications and Quantitative Methods in Archaeology, Prato, 2004) <https://www.ics.forth.gr/isl/publications/paperlink/doerr3_caa 2004.pdf>.

Dooley 2011 — J. Dooley, *Software Development and Professional Practice* (New York, N.Y. : New York: Apress ; Distributed to the trade worldwide by Springer, 2011).

Eide and others 2011 — O. Eide and others, 'Encoding Cultural Heritage Information for the Semantic Web. Procedures for Data Integration through CIDOC-CRM Mapping', in *EPOCH Conference on Open Digital Cultural Heritage Systems Conference: Congresso Rospigliosi, Rome 25-26 February 2008* (presented at the EPOCH Conference on Open Digital Cultural Heritage Systems Conference: Congresso Rospigliosi, Budapest: Archeolingua, 2011) <http://public-repository.epoch-net.org/rome/05%20Procedures%20Data%20Integration.pdf>.

ERA — http://ec.europa.eu/research/era/index_en.htm <last visited: 04.10.2015>.

ERIC 01 — European Commission and Directorate General for Research, *Legal Framework for a European Research Infrastructure Consortium - ERIC: Practical Guidelines* (Luxembourg: EUR-OP, 2010) <https://ec.europa.eu/research/infrastructures/pdf/eric_en.pdf>.

ERIC 02 — European Commission and Directorate General for Research, *ERIC Practical Guidelines: Legal Framework for a European Research Infrastructure Consortium.* (Luxembourg: Publications Office, 2015) <http://dx.publications.europa.eu/10.2777/72348>.

ERIC 03 — https://ec.europa.eu/research/infrastructures/index_en.cfm?pg=eric <last visited: 07.10.2015>.

ESFRI 01 — http://ec.europa.eu/research/infrastructures/index_en.cfm?pg=esfri <last visited: 04.10.2015>.

ESFRI 02 — http://ec.europa.eu/research/infrastructures/index_en.cfm?pg=esfri-roadmap <last visited: 04.10.2015>.

ESFRI 03 — European Strategy Forum on Research Infrastructures, *European Research Infrastructures with Global Impact* (Brussels 2012). <http://www.copori.eu/_media/ESFRI_Brochure_210912_lowres.pdf>.

Flanagan 2011 — D. Flanagan, *JavaScript: The Definitive Guide*, 6th ed (Beijing ; Sebastopol, CA: O'Reilly, 2011).

Gallo – Roberto 2012 — P. Gallo and V. Roberto, 'ANTEO: Sharing Archaeological Data on a WebGIS Platform', in *Proceedings of the 2nd Workshop on The New Technologies for Aquileia* (presented at the CEUR Workshop, Aquileia, 2012), CMXLVIII <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.416.9264&rep=rep1&type=pdf>.

Geser – Niccolucci 2013 — G. Geser and F. Niccolucci, 'Virtual Museums, Digital Reference Collections and E-Science Environments', *Uncommon Culture*, 3.5/6 (2013).

Gilissen 2013 — V. Gilissen, 'Past the Opening: Building towards the Present, on-Going Dissemination of Dutch Archaeological Data as Part of the DANS Archive', in *Conference Proceedings Opening the Past 2013: Archaeology of the Future* (presented at Opening the Past 2013: Archaeology of the Future, Pisa, 2013) <http://depot.knaw.nl/14882/1/Past_the_Opening-_building_towards_the_ present,_on-going_dissemination_of_Dutch_archaeological_data_as_part_of_the_DANS_ archive.pdf>.

Gough – Stallman 2005	B. Gough and R. M. Stallman, *An Introduction to GCC: For the GNU Compilers Gcc and G++*, A Network Theory Manual, 2. print. and updated (Bristol: Network Theory Ltd, 2005).
von Groote-Bidlingmaier and others 2015	C. von Groote-Bidlingmaier and others, 'Interactive WebGIS for Archaeological Settlement Pattern Analysis–A Requirement Analysis', in *27th International Cartographic Conference [and] 16th General Assembly August 23-28, 2015, Rio de Janeiro, Brazil* (presented at the 27th International Cartographic Conference, Rio de Janeiro, 2015) <http://icaci.org/files/documents/ICC_proceedings /ICC2015/papers/33/fullpaper/T33-1009_1428677 938.pdf>.
Henderson 2014	C. Henderson, *Mastering Geoserver: A Holistic Guide to Implementing a Robust, Scalable, and Secure Enterprise Geospatial Data Hosting System by Leveraging the Power of GeoServer* (Birmingham: Packt Publ, 2014).
HERA	S. Kaur-Pedersen and G. Kladakis, *The HERA Survey on Infrastructural Research Facilities and Practices for the Humanities in Europe*, 2006 <https://heranet.info/system/files/ HERAJRPdocuments/Deliverables/d7.1.1_hera_survey_on_humanities_infrastructures.pdf>.
Herban – Vilceanu 2013	S. Herban and B. Vilceanu, 'TLS and Digital Photogrammetry Studies in Cultural Heritage Domain', in *Recent Advances in Geodesy and Geomatics Engineering: Proceedings of the 1st European Conference of Geodesy and Geomatics Engineering (GENG '13)* (presented at the 1st European Conference of Geodesy and Geomatics Engineering, Antalya, 2013) <http://www.gf.uns.ac.rs/~zbornik/doc/NS2015.097.pdf>.
Hollosi 2012	A. Hollosi, *Von Geodaten bis NoSQL: leistungsstarke PHP-Anwendungen ; aktuelle Techniken und Methoden für Fortgeschrittene ; [MySQL 5.6, PostGIS 2.0, Redis 2.4, MongoDB 2.0, Varnish 3.0, Memcached 1.4]* (München: Hanser, 2012).
Horton 2014	I. Horton, *Beginning C++*, The Expert's Voice in C++ (California: Apress, 2014).
Iacovella – Youngblood 2013	S. Iacovella and B. Youngblood, *GeoServer Beginner's Guide: Share and Edit Geospatial Data with This Open Source Software Server*, 1. publ (Birmingham: Packt Publ, 2013).
Isaksen and others 2009	L. Isaksen and others, 'Linking Archaeological Data', in *Making History Interactive. Proceedings of the 37th International Conference on Computer Applications and Quantitative Methods in Archaeology* (presented at the Making History Interactive. 37th International Conference on Computer Applications and Quantitative Methods in Archaeology, Williamsburg, 2009) <http://eprints.ecs.soton.ac.uk/18240>.
Josuttis 2012	N. M. Josuttis, *The C++ Standard Library: A Tutorial and Reference*, 2nd ed (Upper Saddle River, NJ: Addison-Wesley, 2012).
Karagiannis and others 2012	K. N Vavliakis and others, 'Semantic Web in Cultural Heritage after 2020', in *The Semantic Web -- ISWC 2012* (presented at the 11th International Semantic Web Conference 2012, Boston, 2012), XI <http://issel.ee.auth.gr/wp-content/uploads/publications/sw2022_paper10.pdf>.
Kleuker 2011	S. Kleuker, *Grundkurs Datenbankentwicklung: von der Anforderungsanalyse zur komplexen Datenbankanfrage*, Studium, 2. Aufl (Wiesbaden: Vieweg + Teubner, 2011).
Mantegari – De Salvo 2008	G. Mantegari and M. De Salvo, 'Free and Open Source WebGIS Solutions for the PO-BASyN Project', in *Proceedings of the 36th CAA Conference* (presented at the CAA 2008, Budapest, 2008), <http://proceedings.caaconference.org/files/2008/CD50_Mantegari_DeSalvo_ CAA2008.pdf>.
Matsuda – Lea 2013	K. Matsuda and R. Lea, *WebGL Programming Guide: Interactive 3D Graphics Programming with WebGL*, Always Learning (Upper Saddle River, NJ: Addison-Wesley, 2013).
May and others 2011	K. May and others, 'Semantic Technologies Enhancing Links and Linked Data for Archaeological Resources', in *Revive the Past. Computer Applications and Quantitative Methods in Archaeology (CAA). Proceedings of the 39th International Conference* (presented at the CAA, Beijing, 2011) <http://proceedings.caaconference.org/files/2011/29_May_et_al_CAA2011.pdf>.
Meyer and others 2006	E. Meyer and others, 'Integration of Heterogeneous Cultural Heritage Data in a Web-Based Information System: A Case Study from Vianden Castle, Luxembourg', in *Digital Discovery. Exploring New Frontiers in Human Heritage. Proceedings of the 34th International Conference on Computer Applications and Quantitative Methods in Archaeology* (presented at 34th Annual Conference on Computer Applications and Quantitative Methods in Archaeology, Fargo, 2006)

	<https://halshs.archives-ouvertes.fr/halshs-00280680>.
Möller 2001	C. Möller, 'Bodenkundliche Feldmethoden in Der Archäologie', *Archäologisches Korrespondenzblatt*, 6.1 (2001) <http://www.grabung-ev.de/grabung_aktuell/GA14-03.pdf>.
Nedelcu 2013	C. Nedelcu, *Nginx HTTP Server: Adopt Nginx for Your Web Applications to Make the Most of Your Infrastructure and Serve Pages Faster than Ever*, 2. Aufl (Birmingham: Packt Publ, 2013).
Niccolucci – Richards 2013	F. Niccolucci and J. D Richards, 'ARIADNE: Advanced Research Infrastructures for Archaeological Dataset Networking in Europe', *International Journal of Humanities and Arts Computing*, 7.1-2 (2013) <http://e-a-a.org/TEA/archive/TEA_39_SUMMER_2013/rep1_39.pdf>.
Niven - Wright 2014	K. Niven and H. Wright, *Initial Report on Good Practices*, 2014 <http://www.ariadne-infrastructure.eu/content/download/4250/24576/file/ARIADNE_D4.4._Initial%20_Report_Good_Practices.pdf>.
Nixon 2014	R. Nixon, *Learning PHP, MySQL, JavaScript, CSS & HTML5: A Step-by-Step Guide to Creating Dynamic Websites* (O'Reilly Media, Inc., 2014).
Owens 2006	M. Owens, *The Definitive Guide to SQLite* (New York: Apress, 2006).
Parisi 2014	T. Parisi, *Programming 3D Applications with HTML5 and WebGL*, First edition (Beijing: O'Reilly, 2014).
Platzer – Leonhardt 2005	W. Platzer and H. Leonhardt, *Taschenatlas Anatomie* (Georg Thieme Verlag, 2005), I.
Puhl 2015	J. Puhl and others, 'Diskussion und Definition eines Research Data LifeCycle für die digitalen Geisteswissenschaften', *DARIAH-DE Working Papers*, 11 (2015) <http://webdoc.sub.gwdg.de/pub/mon/dariah-de/dwp-2015-11.pdf>.
Reibold 2013	H. Reibold, *XAMPP 1.8 Kompakt: Das Anwenderhandbuch* (Brain-Media, 2013).
Richards 2012	J. Richards, 'Digital Infrastructures for Archaeological Research: A European Perspective', *CSA Newsletter*, 25.2 (2012) <http://csanet.org/newsletter/fall12/nlf1202.html>.
Richards-Rissetto and others 2010	H. Richards-Rissetto and others, 'Kinect and 3D GIS in Archaeology', in *Anthropology Faculty Publications* (presented at the 18th International Conference on Virtual Systems and Multimedia (VSMM), Milan: IEEE, 2012) <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6365942>.
Riding the Wave 2010	*Riding the Wave. How Europe Can Gain from the Rising Tide of Scientific Data. Final Report of the European High-Level Expert Group on Scientific Data*, A Submission to the European Commission, 2010 <http://cordis.europa.eu/fp7/ict/e-infrastructure/docs/hlg-sdi-report.pdf>.
Scianna – Villa 2011	A. Scianna and B. Villa, 'GIS Applications in Archaeology', *Archeologia E Calcolatori*, 22 (2011) <http://eprints.bice.rm.cnr.it/4245/>.
Sebillo and others 2003	M. Sebillo and others, 'A WebGIS for Promoting Archaeological Assets', *Archives and Museum Informatics*, 2003 <http://www.archimuse.com/publishing/ichim03/068C.pdf>.
Semeraro 2007	G. Semeraro, 'LandLab Project and Archaeology On-Line. Web-Based Systems for the Study of Settlement Patterns and Excavation Data in Classical Archaeology', *Archeologia e Calcolatori*, 18 (2007) <http://eprints.bice.rm.cnr.it/1076/>.
Stroustrup 2013	B. Stroustrup, *The C++ Programming Language*, Fourth edition (Upper Saddle River, NJ: Addison-Wesley, 2013).
STELLAR	http://hypermedia.research.southwales.ac.uk/kos/stellar/ <last visited: 16.10.2015>.
Szász and others 2006	B. Szász and others, 'Cultural Heritage on the Semantic Web – the Museum24 Project', in *Digital Semantic Content across Cultures Conference* (presented at the Digital Semantic Content across Cultures Conference, Paris, 2006) <http://www.seco.tkk.fi/events/2006/2006-05-04-websemantique/presentations/articles/Szasz-museum24Paris.pdf>.
Tatroe and others 2013	K. Tatroe, Peter MacIntyre, and Rasmus Lerdorf, *Programming Php* (O'Reilly Media, Inc., 2013).
ThreeJS 2013	J. Dirksen, *Learning Three.Js: The JavaScript 3D Library for WebGL: Create and Animate Stunning 3D Graphics Using the Open Source Three.js JavaScript Library* (Birmingham: Packt Publishing

Limited, 2013).

Tsalgatidou –Pilioura 2002 — A. Tsalgatidou and T. Pilioura, ‘An Overview of Standards and Related Technology in Web Services’, *Distributed and Parallel Databases*, 12.2–3 (2002).

UML Glasklar 2012 — C. Rupp, S. Queins and die SOPHISTEN, *UML 2 Glasklar* (München: Carl Hanser Verlag, 2012).

Unsworth 2006 — J. Unsworth, *Our Cultural Commonwealth: The Report of the American Council of Learned Societies Commission on Cyberinfrastructure for the Humanities and Social Sciences* (ACLS: New York, 2006) <https://www.ideals.illinois.edu/handle/2142/189>.

Van der Vlist 2002 — E. Van der Vlist, *XML Schema. The W3C’s Object-Oriented Descriptions for XML*, 1. ed (O’Reilly, 2002) <http://ebooks.elportal.info/O'Reilly%20-%20Xml%20Schema,%201st%20 Edition.pdf>.

Vonhoegen 2013 — H. Vonhoegen, *XML für Eilige* (Köln: Helmut Vonhoegen, 2013) <http://nbn-resolving.de/urn:nbn:de:101:1-201309042794>.

Williams 2012 — A. Williams, *C++ Concurrency in Action: Practical Multithreading* (Shelter Island, NY: Manning, 2012).

APPENDIX

DATA-MODEL

```
<?xml version="1.0" encoding="UTF-8"?>
<!--  Author: Mag. Thomas Winklehner
  Purpose/Description: Datastructure/-model in Accordance with Schema-Document
  Date: 21.11.2015
  Warning: DO ONLY MODIFY THIS DOCUMENT WITH CARE
-->
<Database xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.thomaswinklehner.at/ArchaeologicalDatabase"
xsi:schemaLocation="http://www.thomaswinklehner.at/ArchaeologicalDatabase DatabaseStructureSchema.xsd">

 <Table TableName="KatastGem" Use="Vocabulary">
   <Column ColumnName="ID" TableName="KatastGem">
   <DataColumn>
    <DataColumnType><NUMERIC Precision="05" Scale="00"/></DataColumnType>
    <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
    <ColumnDescriptionGerman>Kennzahl der Katastralgemeinde</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Numeric Identifier of the so called Katastralgemeinde</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P87 is identified by</OntologyPredicate>
   <OntologyEntity>E44 Place Appellation</OntologyEntity>
  </Column>
   <Column ColumnName="Name" TableName="KatastGem">
   <DataColumn>
    <DataColumnType><NVARCHAR Length="035"/></DataColumnType>
    <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
    <ColumnDescriptionGerman>Name der Katastralgemeinde</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Name of the so called Katastralgemeinde</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P87 is identified by</OntologyPredicate>
   <OntologyEntity>E48 Place Name</OntologyEntity>
  </Column>
  <TableDescriptionGerman>Liste der Katastralgemeinden Österreichs</TableDescriptionGerman>
  <TableDescriptionEnglish>Liste of the so called Katastralgemeinden in Austria</TableDescriptionEnglish>
  <OntologyEntity>E53 Place</OntologyEntity>
 </Table>

 <Table TableName="Project" Use="Storage">
   <Column ColumnName="ProjectID" TableName="Project">
   <DataColumn>
    <DataColumnType><NVARCHAR Length="030"/></DataColumnType>
    <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
    <ColumnDescriptionGerman>Projectbezeichner (Primärschlüssel)</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Project ID (Primary Key)</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P1 is identified by</OntologyPredicate>
   <OntologyEntity>E42 Identifier</OntologyEntity>
  </Column>
   <Column ColumnName="KatastGem" TableName="Project">
   <DataColumn>
    <DataColumnType><NUMERIC Precision="05" Scale="00"/></DataColumnType>
    <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
    <Constraint><FOREIGNKEY RefTableName="KatastGem" RefColumnName="ID" DeleteAction="RESTRICT" UpdateAction="CASCADE"/></Constraint>
    <ColumnDescriptionGerman>Foreign Key zur Katastralgemeinde und als Primary Key Teil der Maßnahmennummer</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Foreign Key to Katastralgemeinde and as being Primary Key Part of the Measure
         Number</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P7 took place at</OntologyPredicate>
   <OntologyEntity>E53 Place</OntologyEntity>
  </Column>
   <Column ColumnName="MeasNr" TableName="Project">
   <DataColumn>
    <DataColumnType><NUMERIC Precision="04" Scale="00"/></DataColumnType>
    <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
    <ColumnDescriptionGerman>Kennzahl und Teil von Primary Key und Maßnahmennummer</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>ID is Part of Primary Key and the Measure Number</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P1 is identified by</OntologyPredicate>
   <OntologyEntity>E42 Identifier</OntologyEntity>
  </Column>
   <Column ColumnName="DateBegin" TableName="Project">
   <DataColumn>
    <DataColumnType><OTHER Type="DATE"/></DataColumnType>
    <ColumnDescriptionGerman>Datum des Projektbeginns (JJJJ-MM-DD)</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Date of Projectbegin (JJJJ-MM-DD)</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P4 has Time-Span - E52 Time-Span - P79 beginning is qualified by</OntologyPredicate>
   <OntologyEntity>E62 String</OntologyEntity>
  </Column>
   <Column ColumnName="DateEnd" TableName="Project">
   <DataColumn>
    <DataColumnType><OTHER Type="DATE"/></DataColumnType>
    <ColumnDescriptionGerman>Datum des Projektendes (JJJJ-MM-DD)</ColumnDescriptionGerman>
```

```
    <ColumnDescriptionEnglish>Date of Projectend (JJJJ-MM-DD)</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P4 has Time-Span - E52 Time-Span - P79 end is qualified by</OntologyPredicate>
   <OntologyEntity>E62 String</OntologyEntity>
  </Column>
   <Column ColumnName="MeasAppell" TableName="Project">
   <DataColumn>
    <DataColumnType><NVARCHAR Length="200"/></DataColumnType>
    <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
    <Constraint><SimpleConstraint Type="UNIQUE"/></Constraint>
    <ColumnDescriptionGerman>Offizielle Benennung der Archaeologischen Maßnahme</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Official Appellation of the Archaeologicl Measure</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P1 is identified by</OntologyPredicate>
   <OntologyEntity>E42 Identifier</OntologyEntity>
  </Column>
   <Column ColumnName="FindStorage" TableName="Project">
   <DataColumn>
    <DataColumnType><NVARCHAR Length="200"/></DataColumnType>
    <ColumnDescriptionGerman>Ort der Verwahrung der Funde</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Place of Finds' Storage</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P134 was continued by - E10 Transfer of Custody - P29 custody received by</OntologyPredicate>
   <OntologyEntity>E39 Actor</OntologyEntity>
  </Column>
  <TableDescriptionGerman>Beschreibung des Ausgrabungsprojects</TableDescriptionGerman>
  <TableDescriptionEnglish>Description of the Excavation Project</TableDescriptionEnglish>
  <OntologyEntity>E7 Activity</OntologyEntity>
 </Table>

 <Table TableName="Roles" Use="Vocabulary">
   <Column ColumnName="RoleID" TableName="Roles">
   <DataColumn>
    <DataColumnType><NUMERIC Precision="03" Scale="00"/></DataColumnType>
    <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
    <ColumnDescriptionGerman>Kennnummer einer Projektrolle bzw. -aufgabe</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>ID of a Projectrole or -function</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P1 is identified by</OntologyPredicate>
   <OntologyEntity>E42 Identifier</OntologyEntity>
  </Column>
   <Column ColumnName="Name" TableName="Roles">
   <DataColumn>
    <DataColumnType><NVARCHAR Length="030"/></DataColumnType>
    <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
    <Constraint><SimpleConstraint Type="UNIQUE"/></Constraint>
    <ColumnDescriptionGerman>Name einer Projektrolle bzw. -aufgabe</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Name of a Projectrole or -function</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P1 is identified by</OntologyPredicate>
   <OntologyEntity>E42 Identifier</OntologyEntity>
  </Column>
  <TableDescriptionGerman>Liste der Projektrollen</TableDescriptionGerman>
  <TableDescriptionEnglish>List of Projectroles</TableDescriptionEnglish>
  <OntologyEntity>E55 Type</OntologyEntity>
 </Table>

 <Table TableName="Person" Use="Storage">
   <Column ColumnName="ID" TableName="Person">
   <DataColumn>
    <DataColumnType><NUMERIC Precision="05" Scale="00"/></DataColumnType>
    <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
    <ColumnDescriptionGerman>Kennnummer einer Person</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>ID of a Person</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P131 is identified by</OntologyPredicate>
   <OntologyEntity>E82 Actor Apellation</OntologyEntity>
  </Column>
   <Column ColumnName="FirstName" TableName="Person">
   <DataColumn>
    <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
    <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
    <ColumnDescriptionGerman>Vorname einer Person</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>First Name of a Person</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P131 is identified by</OntologyPredicate>
   <OntologyEntity>E82 Actor Apellation</OntologyEntity>
  </Column>
   <Column ColumnName="LastName" TableName="Person">
   <DataColumn>
    <DataColumnType><NVARCHAR Length="030"/></DataColumnType>
    <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
    <ColumnDescriptionGerman>Nachname einer Person</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Last Name of a Person</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P131 is identified by</OntologyPredicate>
   <OntologyEntity>E82 Actor Apellation</OntologyEntity>
  </Column>
  <TableDescriptionGerman>Liste der beteiligen Personen</TableDescriptionGerman>
```

```
 <TableDescriptionEnglish>List of People involved</TableDescriptionEnglish>
 <OntologyEntity>E21 Person</OntologyEntity>
</Table>

<Table TableName="Role" Use="Storage">
  <Column ColumnName="PersonID" TableName="Role">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="05" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="Person" RefColumnName="ID" DeleteAction="CASCADE" UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Foreign Key zur Person</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Foreign Key to Person</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyEntity>E21 Person</OntologyEntity>
 </Column>
  <Column ColumnName="RoleID" TableName="Role">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="03" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="Roles" RefColumnName="RoleID" DeleteAction="SET NULL" UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Foreign Key zur Kennnummer einer Projektrolle bzw. -aufgabe</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Foreign Key to ID of a Projectrole or -function</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P14.1 in the role of</OntologyPredicate>
  <OntologyEntity>E55 Type</OntologyEntity>
 </Column>
  <Column ColumnName="ProjectID" TableName="Role">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="030"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="Project" RefColumnName="ProjectID" DeleteAction="CASCADE"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Projektbezeichner</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Project ID</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyEntity>E7 Activity</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Verknüpfung von Person, Rolle und Project</TableDescriptionGerman>
 <TableDescriptionEnglish>Linkage between Person, Role and Project</TableDescriptionEnglish>
 <OntologyPredicate>P14 carried out by (performed)</OntologyPredicate>
</Table>

<Table TableName="Site" Use="Storage">
  <Column ColumnName="ProjectID" TableName="Site">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="030"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="Project" RefColumnName="ProjectID" DeleteAction="CASCADE"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Foreign Key zum Projekt</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Foreign Key to Project</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P8 witnessed</OntologyPredicate>
  <OntologyEntity>E7 Activity</OntologyEntity>
 </Column>
  <Column ColumnName="SiteID" TableName="Site">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <ColumnDescriptionGerman>Kennzahl des Untersuchungsgebiets</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>ID of the Site</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified</OntologyPredicate>
  <OntologyEntity>E42 Identifier</OntologyEntity>
 </Column>
 <Column ColumnName="Location" TableName="Site">
  <SpatialColumn Dimension="XYZ" Nullable="1">POLYGON</SpatialColumn>
  <OntologyPredicate>P156 occupies - E53 Place - 168 place is defined by</OntologyPredicate>
  <OntologyEntity>E94 Space Primitive</OntologyEntity>
 </Column>
 <Column ColumnName="PlaceAddress" TableName="Site">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="200"/></DataColumnType>
   <ColumnDescriptionGerman>Addresse oder Flurbezeichnung</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Address or Sitename</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P156 occupies - E53 Place - 168 place is defined by - Location [E94 Space Primitive] - P87 is identified by
  </OntologyPredicate>
  <OntologyEntity>E45 Address</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Grabungsstellen (Quadranten, Schnitte,...)</TableDescriptionGerman>
 <TableDescriptionEnglish>Areas of excavational Interest</TableDescriptionEnglish>
 <OntologyEntity>E27 Site</OntologyEntity>
</Table>

<Table TableName="ObjectGroup" Use="Storage">
```

```
  <Column ColumnName="ObjectGroupNr" TableName="ObjectGroup">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="05" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <ColumnDescriptionGerman>Kennnummer der ObjektGruppe</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>ID of the ObjectGroup</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E42 Identifier</OntologyEntity>
 </Column>
  <Column ColumnName="ObjectGroupTerm" TableName="ObjectGroup">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="100"/></DataColumnType>
   <Constraint><SimpleConstraint Type="UNIQUE"/></Constraint>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <ColumnDescriptionGerman>Bezeichnung der ObjektsGruppe</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Name of the ObjectGroup</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E41 Appellation</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Objekt Gruppen sind Gruppen von Objekten</TableDescriptionGerman>
 <TableDescriptionEnglish>ObjectGroups are Groups of Objects</TableDescriptionEnglish>
 <OntologyEntity>E24 Physical Man-Made Thing</OntologyEntity>
</Table>

<Table TableName="Object" Use="Storage">
  <Column ColumnName="ObjectNr" TableName="Object">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="05" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <ColumnDescriptionGerman>Kennnummer des Objekts</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>ID of the Object</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E42 Identifier</OntologyEntity>
 </Column>
  <Column ColumnName="ObjectTerm" TableName="Object">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="100"/></DataColumnType>
   <Constraint><SimpleConstraint Type="UNIQUE"/></Constraint>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <ColumnDescriptionGerman>Bezeichnung des Objekts</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Name of the Object</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E41 Appellation</OntologyEntity>
 </Column>
  <Column ColumnName="ObjectGroupNr" TableName="Object">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="05" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="ObjectGroup" RefColumnName="ObjectGroupNr" DeleteAction="RESTRICT"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>zugehörige ObjektGruppe</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>corresponding ObjectGroup</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P46 forms part of</OntologyPredicate>
  <OntologyEntity>E24 Physical Man-Made Thing</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Objekte sind Gruppen von Stratifikationseinheiten</TableDescriptionGerman>
 <TableDescriptionEnglish>Objects are Groups of Units of Stratification</TableDescriptionEnglish>
 <OntologyEntity>E24 Physical Man-Made Thing</OntologyEntity>
</Table>

<Table TableName="UnitOfStratification" Use="Storage">
  <Column ColumnName="USNr" TableName="UnitOfStratification">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="06" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <ColumnDescriptionGerman>Kennnummer der Stratifikationseinheit</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>ID of the Unit of Stratification</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E42 Identifier</OntologyEntity>
 </Column>
  <Column ColumnName="USTerm" TableName="UnitOfStratification">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="200"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <ColumnDescriptionGerman>Verbale Bezeichnung der Statifikationseinheit</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Verbal Appellation of the Unit of Stratification</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E41 Appellation</OntologyEntity>
 </Column>
  <Column ColumnName="ProjectID" TableName="UnitOfStratification">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="030"/></DataColumnType>
```

```
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="Project" RefColumnName="ProjectID" DeleteAction="CASCADE"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Foreign Key zur Katastralgemeinde und als Primary Key Teil der Maßnahmennummer</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Foreign Key to Katastralgemeinde and as being Primary Key Part of the Measure Number
   </ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P12 was present at</OntologyPredicate>
  <OntologyEntity>E7 Activity</OntologyEntity>
 </Column>
  <Column ColumnName="Operator" TableName="UnitOfStratification">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="05" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="Person" RefColumnName="ID" DeleteAction="NO ACTION" UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Kennnummer einer Person</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>ID of a Person</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P140 was attributed by - E13 Attribute Assignment - P14 carried out by</OntologyPredicate>
  <OntologyEntity>E21 Person</OntologyEntity>
 </Column>
  <Column ColumnName="RecordingDate" TableName="UnitOfStratification">
  <DataColumn>
   <DataColumnType><OTHER Type="DATE"/></DataColumnType>
   <ColumnDescriptionGerman>Datum der Datenaufzeichnung (JJJJ-MM-DD)</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Date of Datarecord (JJJJ-MM-DD)</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P140 was attributed by - E13 Attribute Assignment - P4 has time-span - E52 Time-Span - P78 is identified by
  </OntologyPredicate>
  <OntologyEntity>E50 Date</OntologyEntity>
 </Column>
  <Column ColumnName="Interpretation" TableName="UnitOfStratification">
  <DataColumn>
   <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
   <ColumnDescriptionGerman>Allgemeinde Beschreibung</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>General Description</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P3 has note</OntologyPredicate>
  <OntologyEntity>E62 String</OntologyEntity>
 </Column>
  <Column ColumnName="ObjectNr" TableName="UnitOfStratification">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="05" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="Object" RefColumnName="ObjectNr" DeleteAction="RESTRICT"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>zugehöriges Objekt</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>corresponding Object</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P3 has note</OntologyPredicate>
  <OntologyEntity>E62 String</OntologyEntity>
 </Column>
  <Column ColumnName="LinkToModel" TableName="UnitOfStratification">
  <DataColumn>
   <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
   <ColumnDescriptionGerman>Link zu Modell</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Link to Model</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P138 has representation</OntologyPredicate>
  <OntologyEntity>E38 Image</OntologyEntity>
 </Column>
  <Column ColumnName="DrawingSVG" TableName="UnitOfStratification">
  <DataColumn>
   <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
   <ColumnDescriptionGerman>SVG Zeichnung</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>SVG Drawing</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P138 has representation</OntologyPredicate>
  <OntologyEntity>E38 Image</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Stratifikationseinheit allgemein</TableDescriptionGerman>
 <TableDescriptionEnglish>Unit of Stratification general</TableDescriptionEnglish>
 <OntologyEntity>E18 Physical Thing</OntologyEntity>
</Table>

<Table TableName="Stratigraphy" Use="Storage">
  <Column ColumnName="US" TableName="Stratigraphy">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="06" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="UnitOfStratification" RefColumnName="USNr" DeleteAction="CASCADE"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Kennnummer der Stratifikationseinheit</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>ID of the Unit of Stratification</ColumnDescriptionEnglish>
  </DataColumn>
 </Column>
  <Column ColumnName="USolder" TableName="Stratigraphy">
  <DataColumn>
```

```
   <DataColumnType><NUMERIC Precision="06" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="UnitOfStratification" RefColumnName="USNr" DeleteAction="CASCADE"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>zeitlich frühere Stratifikationseinheit</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>preceeding Unit of Stratification</ColumnDescriptionEnglish>
  </DataColumn>
 </Column>
 <TableDescriptionGerman>Referenzierung von US mit der/einer vorhergehenden US</TableDescriptionGerman>
 <TableDescriptionEnglish>Reference of US with a/the preceeding US</TableDescriptionEnglish>
 <OntologyPredicate>P92 was brought into existence by - E63 Beginning of Exsistance - P120 occurs after - E18 Physical Thing
 </OntologyPredicate>
</Table>

<Table TableName="Section" Use="Storage">
 <Column ColumnName="SectionID" TableName="Section">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="030"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <ColumnDescriptionGerman>Bezeichnung des Schnitts</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Appellation of the Section</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E42 Identifier</OntologyEntity>
 </Column>
 <Column ColumnName="Location" TableName="Section">
  <SpatialColumn Dimension="XYZ" Nullable="1">POLYGON</SpatialColumn>
  <OntologyPredicate>P53 has former or current location</OntologyPredicate>
  <OntologyEntity>E53 Place</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Grabungsschnitt</TableDescriptionGerman>
 <TableDescriptionEnglish>Section of the Excavation-Site</TableDescriptionEnglish>
 <OntologyEntity>E27 Site</OntologyEntity>
</Table>

<Table TableName="USToSection" Use="Storage">
  <Column ColumnName="USNr" TableName="USToSection">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="06" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="UnitOfStratification" RefColumnName="USNr" DeleteAction="CASCADE"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Kennnummer der Stratifikationseinheit</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>ID of the Unit of Stratification</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyEntity>E18 Physical Thing</OntologyEntity>
 </Column>
  <Column ColumnName="SectionID" TableName="USToSection">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="030"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="Section" RefColumnName="SectionID" DeleteAction="CASCADE"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Bezeichnung des Schnitts</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Appellation of the Section</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyEntity>E27 Site</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Referenztabelle von Schnitt und Stratifikationseinheit</TableDescriptionGerman>
 <TableDescriptionEnglish>Reference Table of Section and Unit of Stratification</TableDescriptionEnglish>
 <OntologyPredicate>P46 is composed of (forms part of)</OntologyPredicate>
</Table>

<Table TableName="Profile" Use="Storage">
  <Column ColumnName="SectionID" TableName="Profile">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="030"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="Section" RefColumnName="SectionID" DeleteAction="CASCADE"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Bezeichnung des Schnitts</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Appellation of the Section</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P46 forms part of</OntologyPredicate>
  <OntologyEntity>E27 Site</OntologyEntity>
 </Column>
  <Column ColumnName="ProfileID" TableName="Profile">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="030"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <ColumnDescriptionGerman>Bezeichnung des Profils</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Appellation of the Profile</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E42 Identifier</OntologyEntity>
```

```
 </Column>
 <Column ColumnName="Location" TableName="Profile">
  <SpatialColumn Dimension="XYZ" Nullable="1">LINESTRING</SpatialColumn>
  <OntologyPredicate>P53 has former or current location</OntologyPredicate>
  <OntologyEntity>E53 Place</OntologyEntity>
 </Column>
 <TableDescriptionGerman>SchnittProfil</TableDescriptionGerman>
 <TableDescriptionEnglish>Profile of the Section</TableDescriptionEnglish>
 <OntologyEntity>E25 Man-Made Feature</OntologyEntity>
</Table>

<Table TableName="USToProfile" Use="Storage">
  <Column ColumnName="USNr" TableName="USToProfile">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="06" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="UnitOfStratification" RefColumnName="USNr" DeleteAction="CASCADE"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Kennnummer der Stratifikationseinheit</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>ID of the Unit of Stratification</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyEntity>E18 Physical Thing</OntologyEntity>
 </Column>
  <Column ColumnName="ProfileID" TableName="USToProfile">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="030"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="Profile" RefColumnName="ProfileID" DeleteAction="CASCADE"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Bezeichnung des Profils</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Appellation of the Profile</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyEntity>E25 Man-Made Feature</OntologyEntity>
 </Column>
  <Column ColumnName="SectionID" TableName="USToProfile">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="030"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="Profile" RefColumnName="SectionID" DeleteAction="CASCADE"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Bezeichnung des Quadranten</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Appellation of the Section</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyEntity>E25 Man-Made Feature</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Referenztabelle von Profil und Stratifikationseinheit</TableDescriptionGerman>
 <TableDescriptionEnglish>Reference Table of Profile and Unit of Stratification</TableDescriptionEnglish>
 <OntologyPredicate>P46 is composed of (forms part of)</OntologyPredicate>
</Table>

<Table TableName="USInterface" Use="Storage">
  <Column ColumnName="USNr" TableName="USInterface">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="06" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="UnitOfStratification" RefColumnName="USNr" DeleteAction="CASCADE"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Kennnummer der Stratifikationseinheit</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>ID of the Unit of Stratification</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E42 Identifier</OntologyEntity>
 </Column>
 <Column ColumnName="Location" TableName="USInterface">
  <SpatialColumn Dimension="XYZ" Nullable="1">POLYGON</SpatialColumn>
  <OntologyPredicate>P156 occupies - E53 Place - 168 place is defined by</OntologyPredicate>
  <OntologyEntity>E94 Space Primitive</OntologyEntity>
 </Column>
 <Column ColumnName="Surface" TableName="USInterface">
  <SpatialColumn Dimension="XYZ" Nullable="1">MULTIPOINT</SpatialColumn>
  <OntologyPredicate>P156 occupies - E53 Place - 168 place is defined by</OntologyPredicate>
  <OntologyEntity>E94 Space Primitive</OntologyEntity>
 </Column>
  <Column ColumnName="PrelimDatingFrom" TableName="USInterface">
  <DataColumn>
   <DataColumnType><OTHER Type="DATE"/></DataColumnType>
   <ColumnDescriptionGerman>Datierung von (JJJJ-MM-DD)</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Dating from (JJJJ-MM-DD)</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P108 was produced by - E12 Production - P4 has time-span - E52 Time-Span - P79 beginning is qualified by
  </OntologyPredicate>
  <OntologyEntity>E62 String</OntologyEntity>
 </Column>
  <Column ColumnName="PrelimDatingTo" TableName="USInterface">
  <DataColumn>
   <DataColumnType><OTHER Type="DATE"/></DataColumnType>
```

```
   <ColumnDescriptionGerman>Datierung bis (JJJJ-MM-DD)</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Dating to (JJJJ-MM-DD)</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P108 was produced by - E12 Production - P4 has time-span - E52 Time-Span - P79 beginning is qualified by
  </OntologyPredicate>
  <OntologyEntity>E62 String</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Stratifikationseinheit als Interface</TableDescriptionGerman>
 <TableDescriptionEnglish>Unit of Stratification as Interface</TableDescriptionEnglish>
 <OntologyEntity>E25 Man-Made Feature</OntologyEntity>
</Table>

<Table TableName="SedimentTexture" Use="Vocabulary">
  <Column ColumnName="Term" TableName="SedimentTexture">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="003"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <ColumnDescriptionGerman>Sediment (Sedimentkürzel nach ÖNORM)</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Sediment (ShortForm of Sediment according ÖNORM)</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E42 Identifier</OntologyEntity>
 </Column>
  <Column ColumnName="Name" TableName="SedimentTexture">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
   <Constraint><SimpleConstraint Type="UNIQUE"/></Constraint>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <ColumnDescriptionGerman>Sediment Ansprache</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Sediment Appellation</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E42 Identifier</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Sediment nach ÖNORM</TableDescriptionGerman>
 <TableDescriptionEnglish>Sediment according ÖNORM</TableDescriptionEnglish>
 <OntologyEntity>E57 Material</OntologyEntity>
</Table>

<Table TableName="Color" Use="Vocabulary">
  <Column ColumnName="Code" TableName="Color">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <ColumnDescriptionGerman>FarbCode</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>ColorCode</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E42 Identifier</OntologyEntity>
 </Column>
  <Column ColumnName="Name" TableName="Color">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="035"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <ColumnDescriptionGerman>Farbbezeichnung</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Color Appellation</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E41 Appellation</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Farbe nach Munsell Soil Color Chart</TableDescriptionGerman>
 <TableDescriptionEnglish>Color According Munsell Soil Color Chart</TableDescriptionEnglish>
 <OntologyEntity>E26 Physical Feature</OntologyEntity>
</Table>

<Table TableName="USSediment" Use="Storage">
  <Column ColumnName="USNr" TableName="USSediment">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="06" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="UnitOfStratification" RefColumnName="USNr" DeleteAction="CASCADE"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Kennnummer der Stratifikationseinheit</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>ID of the Unit of Stratification</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E42 Identifier</OntologyEntity>
 </Column>
  <Column ColumnName="CoarseSoil" TableName="USSediment">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="003"/></DataColumnType>
   <Constraint><FOREIGNKEY RefTableName="SedimentTexture" RefColumnName="Term" DeleteAction="NO ACTION"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Grobsediment (Sedimentkürzel nach ÖNORM)</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Coarse-Sediment (ShortForm of Sediment according ÖNORM)</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P45 consists of</OntologyPredicate>
  <OntologyEntity>E57 Material</OntologyEntity>
 </Column>
```

```
  <Column ColumnName="CoarseSoilPercentage" TableName="USSediment">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="03" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <ColumnDescriptionGerman>Anteil an Grobsediment</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Share of Coarse-Sediment</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P39 was measured by - E16 Measurement - P40 observed dimension - E54 Dimension - P90 has value
  </OntologyPredicate>
  <OntologyEntity>E60 Number</OntologyEntity>
 </Column>
  <Column ColumnName="FineSoil" TableName="USSediment">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="003"/></DataColumnType>
   <Constraint><FOREIGNKEY RefTableName="SedimentTexture" RefColumnName="Term" DeleteAction="NO ACTION"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Feinsediment (Sedimentkürzel nach ÖNORM)</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Fine-Sediment (ShortForm of Sediment according ÖNORM)</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P45 consists of</OntologyPredicate>
  <OntologyEntity>E57 Material</OntologyEntity>
 </Column>
 <Column ColumnName="Color" TableName="USSediment">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
   <Constraint><FOREIGNKEY RefTableName="Color" RefColumnName="Code" DeleteAction="NO ACTION" UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Farbe nach Munsell Soil Color Chart</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Color According Munsell Soil Color Chart</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P56 bears feature</OntologyPredicate>
  <OntologyEntity>E26 Physical Feature</OntologyEntity>
 </Column>
 <Column ColumnName="Location" TableName="USSediment">
  <SpatialColumn Dimension="XYZ" Nullable="1">POLYGON</SpatialColumn>
  <OntologyPredicate>P156 occupies - E53 Place - 168 place is defined by</OntologyPredicate>
  <OntologyEntity>E94 Space Primitive</OntologyEntity>
 </Column>
 <Column ColumnName="UpSurface" TableName="USSediment">
  <SpatialColumn Dimension="XYZ" Nullable="1">MULTIPOINT</SpatialColumn>
  <OntologyPredicate>P56 bears feature - E26 Physical Feature - P156 occupies - E53 Place - 168 place is defined by
  </OntologyPredicate>
  <OntologyEntity>E94 Space Primitive</OntologyEntity>
 </Column>
 <Column ColumnName="DownSurface" TableName="USSediment">
  <SpatialColumn Dimension="XYZ" Nullable="0">MULTIPOINT</SpatialColumn>
  <OntologyPredicate>P56 bears feature - E26 Physical Feature - P156 occupies - E53 Place - 168 place is defined by
  </OntologyPredicate>
  <OntologyEntity>E94 Space Primitive</OntologyEntity>
 </Column>
 <Column ColumnName="PrelimDatingFrom" TableName="USSediment">
  <DataColumn>
   <DataColumnType><OTHER Type="DATE"/></DataColumnType>
   <ColumnDescriptionGerman>Datierung von (JJJJ-MM-DD)</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Dating from (JJJJ-MM-DD)</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P92 was brought into existence by - E63 Beginning of Existence - P4 has time-span - E52 Time-Span - P79
       beginning is qualified by</OntologyPredicate>
  <OntologyEntity>E62 String</OntologyEntity>
 </Column>
  <Column ColumnName="PrelimDatingTo" TableName="USSediment">
  <DataColumn>
   <DataColumnType><OTHER Type="DATE"/></DataColumnType>
   <ColumnDescriptionGerman>Datierung bis (JJJJ-MM-DD)</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Dating to (JJJJ-MM-DD)</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P92 was brought into existence by - E63 Beginning of Existence - P4 has time-span - E52 Time-Span - P79 end is
       qualified by</OntologyPredicate>
  <OntologyEntity>E62 String</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Stratifikationseinheit als Sediment</TableDescriptionGerman>
 <TableDescriptionEnglish>Unit of Stratification as Sediment</TableDescriptionEnglish>
 <OntologyEntity>E19 Physical Object</OntologyEntity>
</Table>

<Table TableName="USArchitectTerm" Use="Vocabulary">
  <Column ColumnName="Part" TableName="USArchitectTerm">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="025"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <ColumnDescriptionGerman>Ansprache des Architekturteils</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Appellation of the architectural Part</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E42 Identifier</OntologyEntity>
 </Column>
  <Column ColumnName="Material" TableName="USArchitectTerm">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
```

```
   <ColumnDescriptionGerman>Holz oder Stein</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Wood or Stone</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P45 consists of</OntologyPredicate>
  <OntologyEntity>E57 Material</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Vocabular für Architekturteile</TableDescriptionGerman>
 <TableDescriptionEnglish>Vocabulary of architectural Parts</TableDescriptionEnglish>
 <OntologyEntity>E22 Man-Made Object</OntologyEntity>
</Table>

<Table TableName="USWoodPart" Use="Storage">
  <Column ColumnName="USNr" TableName="USWoodPart">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="06" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="USWood" RefColumnName="USNr" DeleteAction="CASCADE" UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Stratifikationseinheit</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Unit of Stratification</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P46 forms part of</OntologyPredicate>
  <OntologyEntity>E22 Man-Made Object</OntologyEntity>
 </Column>
  <Column ColumnName="Amount" TableName="USWoodPart">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="04" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <ColumnDescriptionGerman>Anzahl der Teile</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Number of Parts</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P57 has number of parts</OntologyPredicate>
  <OntologyEntity>E60 Number</OntologyEntity>
 </Column>
  <Column ColumnName="Part" TableName="USWoodPart">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="025"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="USArchitectTerm" RefColumnName="Part" DeleteAction="NO ACTION"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Ansprache des Architekturteils</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Appellation of the architectural Part</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P56 is composed of</OntologyPredicate>
  <OntologyEntity>E22 Man-Made Object</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Referenztabelle für Architekturteile und Holzarchitektur</TableDescriptionGerman>
 <TableDescriptionEnglish>Reference Table for architectural Parts and wooden Architecture</TableDescriptionEnglish>
 <OntologyEntity>E22 Man-Made Object</OntologyEntity>
</Table>

<Table TableName="USDimension" Use="Vocabulary">
  <Column ColumnName="Term" TableName="USDimension">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="010"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <ColumnDescriptionGerman>Vokabular</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Vocabulary</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E41 Appellation</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Signifikanz der Längen-, Breiten- und Höhenmessungen</TableDescriptionGerman>
 <TableDescriptionEnglish>Significance of Measurements</TableDescriptionEnglish>
 <OntologyEntity>E55 Type</OntologyEntity>
</Table>

<Table TableName="Orientation" Use="Vocabulary">
  <Column ColumnName="ShortTerm" TableName="Orientation">
  <DataColumn>
   <DataColumnType><NCHAR Length="007"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <ColumnDescriptionGerman>Kürzel der Ausrichtung</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>ShortTerm of the Direction</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E42 Identifier</OntologyEntity>
 </Column>
  <Column ColumnName="LongTerm" TableName="Orientation">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="030"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><SimpleConstraint Type="UNIQUE"/></Constraint>
   <ColumnDescriptionGerman>Ausrichtung</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Direction</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E41 Appellation</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Ausrichtung</TableDescriptionGerman>
```

```
 <TableDescriptionEnglish>Direction</TableDescriptionEnglish>
 <OntologyEntity>E3 Condition State</OntologyEntity>
</Table>

<Table TableName="USWNogging" Use="Vocabulary">
  <Column ColumnName="Material" TableName="USWNogging">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <ColumnDescriptionGerman>Material der Ausfachung</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Material of the Nogging</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E41 Appellation</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Ausfachung</TableDescriptionGerman>
 <TableDescriptionEnglish>Nogging</TableDescriptionEnglish>
 <OntologyEntity>E22 Man-Made Object</OntologyEntity>
</Table>

<Table TableName="USWType" Use="Vocabulary">
  <Column ColumnName="TermTaxonomy" TableName="USWType">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="030"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <ColumnDescriptionGerman>Taxonomie der Baumart</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Taxonomy of the Wood Type</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E42 Identifier</OntologyEntity>
 </Column>
  <Column ColumnName="TermGerman" TableName="USWType">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="030"/></DataColumnType>
   <Constraint><SimpleConstraint Type="UNIQUE"/></Constraint>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <ColumnDescriptionGerman>deutscher Begriff</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>german Term</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E41 Appellation</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Baum- oder Holzart</TableDescriptionGerman>
 <TableDescriptionEnglish>Wood Type</TableDescriptionEnglish>
 <OntologyEntity>E57 Material</OntologyEntity>
</Table>

<Table TableName="USWood" Use="Storage">
  <Column ColumnName="USNr" TableName="USWood">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="06" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="UnitOfStratification" RefColumnName="USNr" DeleteAction="CASCADE"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Kennnummer Stratifikationseinheit</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>ID of the Unit of Stratification</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E42 Identifier</OntologyEntity>
 </Column>
  <Column ColumnName="LengthCM" TableName="USWood">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="05" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <ColumnDescriptionGerman>Länge der Stratifikationseinheit (Holzarchitektur) in Zentimeter</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Length of the of the Unit of Stratification (wooden Architecture) in Centimeter
   </ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P39 was measured by - E16 Measurement - P40 observed dimension - E54 Dimension - P90 has value
  </OntologyPredicate>
  <OntologyEntity>E60 Number</OntologyEntity>
 </Column>
  <Column ColumnName="LengthDimension" TableName="USWood">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="010"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="USDimension" RefColumnName="Term" DeleteAction="RESTRICT"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Signifikanz der Längenangabe</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Significance of the Length Measurement</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P39 was measured by - E16 Measurement - P3.1 has type</OntologyPredicate>
  <OntologyEntity>E55 Type</OntologyEntity>
 </Column>
  <Column ColumnName="WidthCM" TableName="USWood">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="05" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <ColumnDescriptionGerman>Breite der Stratifikationseinheit (Holzarchitektur) in Zentimeter</ColumnDescriptionGerman>
```

```
  <ColumnDescriptionEnglish>Width of the of the Unit of Stratification (wooden Architecture) in Centimeter
  </ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P39 was measured by - E16 Measurement - P40 observed dimension - E54 Dimension - P90 has value
 </OntologyPredicate>
 <OntologyEntity>E60 Number</OntologyEntity>
</Column>
 <Column ColumnName="WidthDimension" TableName="USWood">
 <DataColumn>
  <DataColumnType><NVARCHAR Length="010"/></DataColumnType>
  <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
  <Constraint><FOREIGNKEY RefTableName="USDimension" RefColumnName="Term" DeleteAction="RESTRICT"
       UpdateAction="CASCADE"/></Constraint>
  <ColumnDescriptionGerman>Signifikanz der Breitenangabe</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Significance of the Width Measurement</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P39 was measured by - E16 Measurement - P3.1 has type</OntologyPredicate>
 <OntologyEntity>E55 Type</OntologyEntity>
</Column>
 <Column ColumnName="HeigthCM" TableName="USWood">
 <DataColumn>
  <DataColumnType><NUMERIC Precision="05" Scale="00"/></DataColumnType>
  <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
  <ColumnDescriptionGerman>Höhe der Stratifikationseinheit (Holzarchitektur) in Zentimeter</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Height of the of the Unit of Stratification (wooden Architecture) in Centimeter
  </ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P39 was measured by - E16 Measurement - P40 observed dimension - E54 Dimension - P90 has value
 </OntologyPredicate>
 <OntologyEntity>E60 Number</OntologyEntity>
</Column>
 <Column ColumnName="HeigthDimension" TableName="USWood">
 <DataColumn>
  <DataColumnType><NVARCHAR Length="010"/></DataColumnType>
  <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
  <Constraint><FOREIGNKEY RefTableName="USDimension" RefColumnName="Term" DeleteAction="RESTRICT"
       UpdateAction="CASCADE"/></Constraint>
  <ColumnDescriptionGerman>Signifikanz der Höhenangabe</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Significance of the Height Measurement</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P39 was measured by - E16 Measurement - P3.1 has type</OntologyPredicate>
 <OntologyEntity>E55 Type</OntologyEntity>
</Column>
 <Column ColumnName="Orientation" TableName="USWood">
 <DataColumn>
  <DataColumnType><NCHAR Length="007"/></DataColumnType>
  <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
  <Constraint><FOREIGNKEY RefTableName="Orientation" RefColumnName="ShortTerm" DeleteAction="RESTRICT"
       UpdateAction="CASCADE"/></Constraint>
  <ColumnDescriptionGerman>Kürzel der Ausrichtung</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>ShortTerm of the Direction</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P44 has condition</OntologyPredicate>
 <OntologyEntity>E3 Condition State</OntologyEntity>
</Column>
 <Column ColumnName="Context" TableName="USWood">
 <DataColumn>
  <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
  <ColumnDescriptionGerman>Kontext der Stratifikationseinheit</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Context of the Unit of Stratification</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P3 has note</OntologyPredicate>
 <OntologyEntity>E62 String</OntologyEntity>
</Column>
 <Column ColumnName="WoodType" TableName="USWood">
 <DataColumn>
  <DataColumnType><NVARCHAR Length="030"/></DataColumnType>
  <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
  <Constraint><FOREIGNKEY RefTableName="USWType" RefColumnName="TermTaxonomy" DeleteAction="RESTRICT"
       UpdateAction="CASCADE"/></Constraint>
  <ColumnDescriptionGerman>Holzart</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Type of Wood</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P45 consists of</OntologyPredicate>
 <OntologyEntity>E57 Material</OntologyEntity>
</Column>
 <Column ColumnName="Nogging" TableName="USWood">
 <DataColumn>
  <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
  <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
  <Constraint><FOREIGNKEY RefTableName="USWNogging" RefColumnName="Material" DeleteAction="RESTRICT"
       UpdateAction="CASCADE"/></Constraint>
  <ColumnDescriptionGerman>Material der Ausfachung</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Material of Nogging</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P46 is composed of</OntologyPredicate>
 <OntologyEntity>E22 Man-Made Object</OntologyEntity>
</Column>
 <Column ColumnName="MaterialsOthers" TableName="USWood">
```

```
 <DataColumn>
  <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
  <ColumnDescriptionGerman>weitere Materialien</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>other/further Materials</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P45 consists of</OntologyPredicate>
 <OntologyEntity>E57 Material</OntologyEntity>
</Column>
 <Column ColumnName="SurfaceTreatment" TableName="USWood">
 <DataColumn>
  <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
  <ColumnDescriptionGerman>Oberflächenstaltung</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Surface Treatment</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P3 has note</OntologyPredicate>
 <OntologyEntity>E62 String</OntologyEntity>
</Column>
 <Column ColumnName="Preservation" TableName="USWood">
 <DataColumn>
  <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
  <ColumnDescriptionGerman>(Erhaltungs)Zustand</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>State of Preservation</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P44 has condition</OntologyPredicate>
 <OntologyEntity>E3 Condition State</OntologyEntity>
</Column>
 <Column ColumnName="SecondaryUse" TableName="USWood">
 <DataColumn>
  <DataColumnType><OTHER Type="BOOLEAN"/></DataColumnType>
  <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
  <ColumnDescriptionGerman>in Zweitverwendung</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>in secondary Use</ColumnDescriptionEnglish>
 </DataColumn>
</Column>
 <Column ColumnName="Wane" TableName="USWood">
 <DataColumn>
  <DataColumnType><OTHER Type="BOOLEAN"/></DataColumnType>
  <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
  <ColumnDescriptionGerman>Waldkante</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Wane</ColumnDescriptionEnglish>
 </DataColumn>
</Column>
 <Column ColumnName="SapWood" TableName="USWood">
 <DataColumn>
  <DataColumnType><OTHER Type="BOOLEAN"/></DataColumnType>
  <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
  <ColumnDescriptionGerman>Splintholz</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Sapwood</ColumnDescriptionEnglish>
 </DataColumn>
</Column>
 <Column ColumnName="RipeWood" TableName="USWood">
 <DataColumn>
  <DataColumnType><OTHER Type="BOOLEAN"/></DataColumnType>
  <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
  <ColumnDescriptionGerman>Kernholz</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Ripewood</ColumnDescriptionEnglish>
 </DataColumn>
</Column>
 <Column ColumnName="Position" TableName="USWood">
 <DataColumn>
  <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
  <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
  <ColumnDescriptionGerman>Lage</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Position</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P44 has condition</OntologyPredicate>
 <OntologyEntity>E3 Condition State</OntologyEntity>
</Column>
 <Column ColumnName="Toolmarks" TableName="USWood">
 <DataColumn>
  <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
  <ColumnDescriptionGerman>Bearbeitungsspuren</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Toolmarks</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P56 bears feature</OntologyPredicate>
 <OntologyEntity>E25 Man-Made Feature</OntologyEntity>
</Column>
 <Column ColumnName="MarksInscriptions" TableName="USWood">
 <DataColumn>
  <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
  <ColumnDescriptionGerman>Zeichen und Inschriften</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Marks and Inscriptions</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P65 shows visual item</OntologyPredicate>
 <OntologyEntity>E37 Mark</OntologyEntity>
</Column>
 <Column ColumnName="Joints" TableName="USWood">
 <DataColumn>
  <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
```

```
    <ColumnDescriptionGerman>Verbindungen</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Joints</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P56 bears feature</OntologyPredicate>
   <OntologyEntity>E25 Man-Made Feature</OntologyEntity>
  </Column>
   <Column ColumnName="Relations" TableName="USWood">
   <DataColumn>
    <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
    <ColumnDescriptionGerman>Verbindungen zu anderen Teilen</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Relation to other Parts</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P56 bears feature</OntologyPredicate>
   <OntologyEntity>E25 Man-Made Feature</OntologyEntity>
  </Column>
  <Column ColumnName="Location" TableName="USWood">
   <SpatialColumn Dimension="XYZ" Nullable="1">POLYGON</SpatialColumn>
   <OntologyPredicate>P156 occupies - E53 Place - 168 place is defined by</OntologyPredicate>
   <OntologyEntity>E94 Space Primitive</OntologyEntity>
  </Column>
  <Column ColumnName="UpSurface" TableName="USWood">
   <SpatialColumn Dimension="XYZ" Nullable="1">MULTIPOINT</SpatialColumn>
   <OntologyPredicate>P56 bears feature - E25 Man-Made Feature - P156 occupies - E53 Place - 168 place is defined by
   </OntologyPredicate>
   <OntologyEntity>E94 Space Primitive</OntologyEntity>
  </Column>
  <Column ColumnName="DownSurface" TableName="USWood">
   <SpatialColumn Dimension="XYZ" Nullable="0">MULTIPOINT</SpatialColumn>
   <OntologyPredicate>P56 bears feature - E25 Man-Made Feature - P156 occupies - E53 Place - 168 place is defined by
   </OntologyPredicate>
   <OntologyEntity>E94 Space Primitive</OntologyEntity>
  </Column>
  <Column ColumnName="PrelimProdDatingFrom" TableName="USWood">
   <DataColumn>
    <DataColumnType><OTHER Type="DATE"/></DataColumnType>
    <ColumnDescriptionGerman>Datierung der Herstellung von (JJJJ-MM-DD)</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Dating of Production from (JJJJ-MM-DD)</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P108 was produced by - E12 Production - P4 has time-span - E52 Time-Span - P79 beginning is qualified by
   </OntologyPredicate>
   <OntologyEntity>E62 String</OntologyEntity>
  </Column>
  <Column ColumnName="PrelimProdDatingTo" TableName="USWood">
   <DataColumn>
    <DataColumnType><OTHER Type="DATE"/></DataColumnType>
    <ColumnDescriptionGerman>Datierung der Herstellung bis (JJJJ-MM-DD)</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Dating of Production to (JJJJ-MM-DD)</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P108 was produced by - E12 Production - P4 has time-span - E52 Time-Span - P80 end is qualified by
   </OntologyPredicate>
   <OntologyEntity>E62 String</OntologyEntity>
  </Column>
  <Column ColumnName="PrelimUseDatingTo" TableName="USWood">
   <DataColumn>
    <DataColumnType><OTHER Type="DATE"/></DataColumnType>
    <ColumnDescriptionGerman>Datierung der Verwendung von (JJJJ-MM-DD)</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Dating of Use from (JJJJ-MM-DD)</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P108 was produced by - E12 Production - P4 has time-span - E52 Time-Span - P80 end is qualified by
   </OntologyPredicate>
   <OntologyEntity>E62 String</OntologyEntity>
  </Column>
  <TableDescriptionGerman>Stratificationseinheit als Holzarchitektur</TableDescriptionGerman>
  <TableDescriptionEnglish>Unit of Stratification as wooden Architecture</TableDescriptionEnglish>
  <OntologyEntity>E22 Man-Made Object</OntologyEntity>
 </Table>

 <Table TableName="USBType" Use="Vocabulary">
   <Column ColumnName="Term" TableName="USBType">
   <DataColumn>
    <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
    <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
    <ColumnDescriptionGerman>Aufbau der Mauer im Querschnitt</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Composition of the Wall</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P1 is identified by</OntologyPredicate>
   <OntologyEntity>E41 Appellation</OntologyEntity>
  </Column>
  <TableDescriptionGerman>Aufbau der Mauer im Querschnitt</TableDescriptionGerman>
  <TableDescriptionEnglish>Composition of the Wall</TableDescriptionEnglish>
  <OntologyEntity>E55 Type</OntologyEntity>
 </Table>

 <Table TableName="USBStructure" Use="Vocabulary">
   <Column ColumnName="Term" TableName="USBStructure">
   <DataColumn>
    <DataColumnType><NVARCHAR Length="030"/></DataColumnType>
    <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
    <ColumnDescriptionGerman>Structur des Mauerwerks</ColumnDescriptionGerman>
```

```
   <ColumnDescriptionEnglish>Structure of the Masonry</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E41 Appellation</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Structur des Mauerwerks</TableDescriptionGerman>
 <TableDescriptionEnglish>Structure of the Masonry</TableDescriptionEnglish>
 <OntologyEntity>E29 Design or Procedure</OntologyEntity>
</Table>

<Table TableName="USBuilding" Use="Storage">
  <Column ColumnName="USNr" TableName="USBuilding">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="06" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="UnitOfStratification" RefColumnName="USNr" DeleteAction="CASCADE"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Kennnummer Stratifikationseinheit</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>ID of the Unit of Stratification</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E41 Appellation</OntologyEntity>
 </Column>
  <Column ColumnName="LengthCM" TableName="USBuilding">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="05" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <ColumnDescriptionGerman>Länge der Stratifikationseinheit (Holzarchitektur) in Zentimeter</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Length of the of the Unit of Stratification (wooden Architecture) in Centimeter
   </ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P39 was measured by - E16 Measurement - P40 observed dimension - E54 Dimension - P90 has value
  </OntologyPredicate>
  <OntologyEntity>E60 Number</OntologyEntity>
 </Column>
  <Column ColumnName="LengthDimension" TableName="USBuilding">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="010"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="USDimension" RefColumnName="Term" DeleteAction="RESTRICT"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Signifikanz der Längenangabe</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Significance of the Length Measurement</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P39 was measured by - E16 Measurement - P3.1 has type</OntologyPredicate>
  <OntologyEntity>E55 Type</OntologyEntity>
 </Column>
  <Column ColumnName="WidthCM" TableName="USBuilding">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="05" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <ColumnDescriptionGerman>Breite der Stratifikationseinheit (Holzarchitektur) in Zentimeter</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Width of the of the Unit of Stratification (wooden Architecture) in Centimeter
   </ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P39 was measured by - E16 Measurement - P40 observed dimension - E54 Dimension - P90 has value
  </OntologyPredicate>
  <OntologyEntity>E60 Number</OntologyEntity>
 </Column>
  <Column ColumnName="WidthDimension" TableName="USBuilding">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="010"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="USDimension" RefColumnName="Term" DeleteAction="RESTRICT"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Signifikanz der Breitenangabe</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Significance of the Width Measurement</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P39 was measured by - E16 Measurement - P3.1 has type</OntologyPredicate>
  <OntologyEntity>E55 Type</OntologyEntity>
 </Column>
 <Column ColumnName="HeigthMasonryCM" TableName="USBuilding">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="05" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <ColumnDescriptionGerman>Höhe des Mauerwerks in Zentimeter</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Height of the of the Unit of Stratification (Masonry) in Centimeter</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P39 was measured by - E16 Measurement - P40 observed dimension - E54 Dimension - P90 has value
  </OntologyPredicate>
  <OntologyEntity>E60 Number</OntologyEntity>
 </Column>
 <Column ColumnName="HMDimension" TableName="USBuilding">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="010"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="USDimension" RefColumnName="Term" DeleteAction="RESTRICT"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Signifikanz der Höhenangabe</ColumnDescriptionGerman>
```

```
  <ColumnDescriptionEnglish>Significance of the Height Measurement</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P39 was measured by - E16 Measurement - P3.1 has type</OntologyPredicate>
 <OntologyEntity>E55 Type</OntologyEntity>
</Column>
<Column ColumnName="HeigthFoundingCM" TableName="USBuilding">
 <DataColumn>
  <DataColumnType><NUMERIC Precision="05" Scale="00"/></DataColumnType>
  <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
  <ColumnDescriptionGerman>Höhe des Fundaments in Zentimeter</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Height of the of the Founding in Centimeter</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P39 was measured by - E16 Measurement - P40 observed dimension - E54 Dimension - P90 has value
 </OntologyPredicate>
 <OntologyEntity>E60 Number</OntologyEntity>
</Column>
<Column ColumnName="HFDimension" TableName="USBuilding">
 <DataColumn>
  <DataColumnType><NVARCHAR Length="010"/></DataColumnType>
  <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
  <Constraint><FOREIGNKEY RefTableName="USDimension" RefColumnName="Term" DeleteAction="RESTRICT"
       UpdateAction="CASCADE"/></Constraint>
  <ColumnDescriptionGerman>Signifikanz der Höhenangabe des Fundaments</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Significance of the Height Measurement of the Founding</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P39 was measured by - E16 Measurement - P3.1 has type</OntologyPredicate>
 <OntologyEntity>E55 Type</OntologyEntity>
</Column>
<Column ColumnName="Orientation" TableName="USBuilding">
 <DataColumn>
  <DataColumnType><NCHAR Length="007"/></DataColumnType>
  <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
  <Constraint><FOREIGNKEY RefTableName="Orientation" RefColumnName="ShortTerm" DeleteAction="RESTRICT"
       UpdateAction="CASCADE"/></Constraint>
  <ColumnDescriptionGerman>Kürzel der Ausrichtung</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>ShortTerm of the Direction</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P44 has condition</OntologyPredicate>
 <OntologyEntity>E3 Condition State</OntologyEntity>
</Column>
 <Column ColumnName="Type" TableName="USBuilding">
 <DataColumn>
  <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
  <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
  <Constraint><FOREIGNKEY RefTableName="USBType" RefColumnName="Term" DeleteAction="RESTRICT" UpdateAction="CASCADE"/></Constraint>
  <ColumnDescriptionGerman>Aufbau</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Construction Type</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P108 was produced by - E12 Production - P32 used general technique</OntologyPredicate>
 <OntologyEntity>E55 Type</OntologyEntity>
</Column>
 <Column ColumnName="Structure" TableName="USBuilding">
 <DataColumn>
  <DataColumnType><NVARCHAR Length="030"/></DataColumnType>
  <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
  <Constraint><FOREIGNKEY RefTableName="USBStructure" RefColumnName="Term" DeleteAction="RESTRICT"
       UpdateAction="CASCADE"/></Constraint>
  <ColumnDescriptionGerman>Structur des Mauerwerks</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Structure of the Masonry</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P108 was produced by - E12 Production - P33 used specific technique</OntologyPredicate>
 <OntologyEntity>E29 Design or Procedure</OntologyEntity>
</Column>
 <Column ColumnName="Description" TableName="USBuilding">
 <DataColumn>
  <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
  <ColumnDescriptionGerman>Beschreibung</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Description</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P3 has note</OntologyPredicate>
 <OntologyEntity>E62 String</OntologyEntity>
</Column>
 <Column ColumnName="ButlogHole" TableName="USBuilding">
 <DataColumn>
  <DataColumnType><OTHER Type="BOOLEAN"/></DataColumnType>
  <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
  <ColumnDescriptionGerman>Gerüstloch</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Butlog Hole</ColumnDescriptionEnglish>
 </DataColumn>
</Column>
 <Column ColumnName="BeamHole" TableName="USBuilding">
 <DataColumn>
  <DataColumnType><OTHER Type="BOOLEAN"/></DataColumnType>
  <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
  <ColumnDescriptionGerman>Balkenloch</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Beam Hole</ColumnDescriptionEnglish>
 </DataColumn>
</Column>
 <Column ColumnName="HoleLocation" TableName="USBuilding">
```

```
 <DataColumn>
  <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
  <ColumnDescriptionGerman>Lage und Anordnung der Löcher</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Position of the Holes</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P56 bears feature - E25 Man-Made Feature - P53 has former or current location</OntologyPredicate>
 <OntologyEntity>E53 Place</OntologyEntity>
</Column>
 <Column ColumnName="HoleMeasurements" TableName="USBuilding">
 <DataColumn>
  <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
  <ColumnDescriptionGerman>Beschreibung der Größe der Löcher</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Description of the Size of the Holes</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P56 bears feature - E25 Man-Made Feature - P39 was measured - E16 Measurement</OntologyPredicate>
 <OntologyEntity>E16 Measurement</OntologyEntity>
</Column>
 <Column ColumnName="StyleFeatures" TableName="USBuilding">
 <DataColumn>
  <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
  <ColumnDescriptionGerman>Architekturmerkmale und Stil</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Features of Style and Architecture</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P56 bears feature</OntologyPredicate>
 <OntologyEntity>E25 Man-Made Feature</OntologyEntity>
</Column>
 <Column ColumnName="Relations" TableName="USBuilding">
 <DataColumn>
  <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
  <ColumnDescriptionGerman>Verhältnisse zu anderen Bauteilen</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Relations to other Building Parts</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P56 bears feature</OntologyPredicate>
 <OntologyEntity>E25 Man-Made Feature</OntologyEntity>
</Column>
<Column ColumnName="Location" TableName="USBuilding">
 <SpatialColumn Dimension="XYZ" Nullable="1">POLYGON</SpatialColumn>
 <OntologyPredicate>P156 occupies - E53 Place - 168 place is defined by</OntologyPredicate>
 <OntologyEntity>E94 Space Primitive</OntologyEntity>
</Column>
<Column ColumnName="UpSurface" TableName="USBuilding">
 <SpatialColumn Dimension="XYZ" Nullable="1">MULTIPOINT</SpatialColumn>
 <OntologyPredicate>P56 bears feature - E25 Man-Made Feature - P156 occupies - E53 Place - 168 place is defined by
 </OntologyPredicate>
 <OntologyEntity>E94 Space Primitive</OntologyEntity>
</Column>
<Column ColumnName="DownSurface" TableName="USBuilding"><SpatialColumn Dimension="XYZ"
       Nullable="0">MULTIPOINT</SpatialColumn></Column>
 <Column ColumnName="PrelimProdDatingFrom" TableName="USBuilding">
 <DataColumn>
  <DataColumnType><OTHER Type="DATE"/></DataColumnType>
  <ColumnDescriptionGerman>Datierung der Herstellung von (JJJJ-MM-DD)</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Dating of Production from (JJJJ-MM-DD)</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P56 bears feature - E25 Man-Made Feature - P156 occupies - E53 Place - 168 place is defined by
 </OntologyPredicate>
 <OntologyEntity>E94 Space Primitive</OntologyEntity>
</Column>
 <Column ColumnName="PrelimProdDatingTo" TableName="USBuilding">
 <DataColumn>
  <DataColumnType><OTHER Type="DATE"/></DataColumnType>
  <ColumnDescriptionGerman>Datierung der Herstellung bis (JJJJ-MM-DD)</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Dating of Production to (JJJJ-MM-DD)</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P108 was produced by - E12 Production - P4 has time-span - E52 Time-Span - P79 beginning is qualified by
 </OntologyPredicate>
 <OntologyEntity>E62 String</OntologyEntity>
</Column>
<Column ColumnName="PrelimUseDatingTo" TableName="USBuilding">
 <DataColumn>
  <DataColumnType><OTHER Type="DATE"/></DataColumnType>
  <ColumnDescriptionGerman>Datierung der Verwendung von (JJJJ-MM-DD)</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Dating of Use from (JJJJ-MM-DD)</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P16 was used for - E7 Activity - P4 has time-span - P78 is identified by - E50 Date</OntologyPredicate>
 <OntologyEntity>E62 String</OntologyEntity>
</Column>
 <Column ColumnName="MarksInscriptions" TableName="USBuilding">
 <DataColumn>
  <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
  <ColumnDescriptionGerman>Zeichen und Inschriften</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Marks and Inscriptions</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P65 shows visual item</OntologyPredicate>
 <OntologyEntity>E37 Mark</OntologyEntity>
</Column>
<TableDescriptionGerman>Stratifikationseinheit als Stein- oder Ziegelarchitektur</TableDescriptionGerman>
<TableDescriptionEnglish>Unit of Stratification as Stone- or Brick-Architecture</TableDescriptionEnglish>
<OntologyEntity>E22 Man-Made Object</OntologyEntity>
```

```
</Table>

<Table TableName="USBStoneMaterial" Use="Vocabulary">
  <Column ColumnName="Term" TableName="USBStoneMaterial">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <ColumnDescriptionGerman>Steinmaterial</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Stone Material</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E41 Appellation</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Steinmaterial</TableDescriptionGerman>
 <TableDescriptionEnglish>Stone Material</TableDescriptionEnglish>
 <OntologyEntity>E57 Material</OntologyEntity>
</Table>

<Table TableName="USBElaborationType" Use="Vocabulary">
  <Column ColumnName="Term" TableName="USBElaborationType">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <ColumnDescriptionGerman>Zurichtung der Steine</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Elaboration of the Stones</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E41 Appellation</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Zurichtung der Steine</TableDescriptionGerman>
 <TableDescriptionEnglish>Elaboration of the Stones</TableDescriptionEnglish>
 <OntologyEntity>E29 Design or Procedure</OntologyEntity>
</Table>

<Table TableName="USBComponentStone" Use="Storage">
  <Column ColumnName="USNr" TableName="USBComponentStone">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="06" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="USBuilding" RefColumnName="USNr" DeleteAction="CASCADE"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Kennnummer Stratifikationseinheit</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>ID of the Unit of Stratification</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E42 Identifier</OntologyEntity>
 </Column>
  <Column ColumnName="SizeCM" TableName="USBComponentStone">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="04" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <ColumnDescriptionGerman>Ungefähre durchschnittliche Größe der Steine</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Average Size of the Stones</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P39 was measured by - E16 Measurement - P40 observed dimension - E54 Dimension - P90 has value
  </OntologyPredicate>
  <OntologyEntity>E60 Number</OntologyEntity>
 </Column>
  <Column ColumnName="Material" TableName="USBComponentStone">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="USBStoneMaterial" RefColumnName="Term" DeleteAction="RESTRICT"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Material</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Material</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P45 consists of</OntologyPredicate>
  <OntologyEntity>E57 Material</OntologyEntity>
 </Column>
  <Column ColumnName="Treatment" TableName="USBComponentStone">
  <DataColumn>
   <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
   <ColumnDescriptionGerman>Bearbeitung</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Treatment</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P108 was produced by - E12 Production - P32 used general technique</OntologyPredicate>
  <OntologyEntity>E55 Type</OntologyEntity>
 </Column>
  <Column ColumnName="Spoils" TableName="USBComponentStone">
  <DataColumn>
   <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
   <ColumnDescriptionGerman>verwendete Spolien</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Spoils employed</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P46 is composed of</OntologyPredicate>
  <OntologyEntity>E22 Man-Made Object</OntologyEntity>
 </Column>
  <Column ColumnName="Elaboration" TableName="USBComponentStone">
```

```
  <DataColumn>
   <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="USBElaborationType" RefColumnName="Term" DeleteAction="RESTRICT"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Hauptkomponente</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Main Component</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P108 was produced by - E12 Production - P33 used specific technique</OntologyPredicate>
  <OntologyEntity>E29 Design or Precedure</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Baustoff Stein</TableDescriptionGerman>
 <TableDescriptionEnglish>Building Material Stone</TableDescriptionEnglish>
 <OntologyEntity>E22 Man-Made Object</OntologyEntity>
</Table>

<Table TableName="USBFunctionType" Use="Vocabulary">
  <Column ColumnName="Term" TableName="USBFunctionType">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <ColumnDescriptionGerman>Ziegeltyp</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Brick Type</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E41 Appellation</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Ziegeltyp</TableDescriptionGerman>
 <TableDescriptionEnglish>Brick Type</TableDescriptionEnglish>
 <OntologyEntity>E55 Type</OntologyEntity>
</Table>

<Table TableName="USBFiring" Use="Vocabulary">
  <Column ColumnName="Term" TableName="USBFiring">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <ColumnDescriptionGerman>Lehmziegel oder gebrannter Ziegel</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Loam or fired Brick</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E41 Appellation</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Lehmziegel oder gebrannter Ziegel</TableDescriptionGerman>
 <TableDescriptionEnglish>Loam or fired Brick</TableDescriptionEnglish>
 <OntologyEntity>E55 Type</OntologyEntity>
</Table>

<Table TableName="USBComponentEarthenWare" Use="Storage">
  <Column ColumnName="USNr" TableName="USBComponentEarthenWare">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="06" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="USBuilding" RefColumnName="USNr" DeleteAction="CASCADE"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Kennnummer Stratifikationseinheit</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>ID of the Unit of Stratification</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E42 Identifier</OntologyEntity>
 </Column>
  <Column ColumnName="FunctionType" TableName="USBComponentEarthenWare">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="USBFunctionType" RefColumnName="Term" DeleteAction="RESTRICT"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Ziegeltyp</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Brick Type</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P102.1 has type</OntologyPredicate>
  <OntologyEntity>E55 Type</OntologyEntity>
 </Column>
  <Column ColumnName="Color" TableName="USBComponentEarthenWare">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="Color" RefColumnName="Code" DeleteAction="NO ACTION" UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Farbe nach Munsell Soil Color Chart</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Color According Munsell Soil Color Chart</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P56 bears feature</OntologyPredicate>
  <OntologyEntity>E26 Physical Feature</OntologyEntity>
 </Column>
  <Column ColumnName="ProductionMarks" TableName="USBComponentEarthenWare">
  <DataColumn>
   <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
   <ColumnDescriptionGerman>Herstellungsspuren</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Production Marks and Signs</ColumnDescriptionEnglish>
```

```
  </DataColumn>
  <OntologyPredicate>P65 shows visual item</OntologyPredicate>
  <OntologyEntity>E37 Mark</OntologyEntity>
 </Column>
  <Column ColumnName="Fingerprints" TableName="USBComponentEarthenWare">
  <DataColumn>
   <DataColumnType><OTHER Type="BOOLEAN"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <ColumnDescriptionGerman>Fingerabdrücke</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Fingerprints</ColumnDescriptionEnglish>
  </DataColumn>
 </Column>
  <Column ColumnName="SignsOthers" TableName="USBComponentEarthenWare">
  <DataColumn>
   <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
   <ColumnDescriptionGerman>Andere Spuren</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Other Marks and Signs</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P56 bears feature</OntologyPredicate>
  <OntologyEntity>E25 Man-Made Feature</OntologyEntity>
 </Column>
  <Column ColumnName="Firing" TableName="USBComponentEarthenWare">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="USBFiring" RefColumnName="Term" DeleteAction="RESTRICT"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Lehmziegel oder gebrannter Ziegel</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Loam or fired Brick</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P108 was produced by - E12 Production - P32 used general technique</OntologyPredicate>
  <OntologyEntity>E55 Type</OntologyEntity>
 </Column>
  <Column ColumnName="SharePercentage" TableName="USBComponentEarthenWare">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="03" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <ColumnDescriptionGerman>Anteil von Lehm/Ziegel am Mauerwerk</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Share of Loam/Brick in the Masonry</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P39 was measured by - E16 Measurement - P40 observed dimension - E54 Dimension - P90 has value
  </OntologyPredicate>
  <OntologyEntity>E60 Number</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Baustoff Lehm und Ziegel</TableDescriptionGerman>
 <TableDescriptionEnglish>Building Material Loam and Brick</TableDescriptionEnglish>
 <OntologyEntity>E22 Man-Made Object</OntologyEntity>
</Table>

<Table TableName="USBBSeam" Use="Vocabulary">
  <Column ColumnName="Term" TableName="USBBSeam">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <ColumnDescriptionGerman>Fugenbild</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Seam Elaboration</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E41 Appellation</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Fugenbild</TableDescriptionGerman>
 <TableDescriptionEnglish>Seam Elaboration</TableDescriptionEnglish>
 <OntologyEntity>E22 Man-Made Object</OntologyEntity>
</Table>

<Table TableName="USBBGrainSize" Use="Vocabulary">
  <Column ColumnName="Term" TableName="USBBGrainSize">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <ColumnDescriptionGerman>Partikelgröße</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Grain- or Partikel Size</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E41 Appellation</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Partikelgröße</TableDescriptionGerman>
 <TableDescriptionEnglish>Grain- or Partikel Size</TableDescriptionEnglish>
 <OntologyEntity>E60 Number</OntologyEntity>
</Table>

<Table TableName="USBBHumidity" Use="Vocabulary">
  <Column ColumnName="Term" TableName="USBBHumidity">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <ColumnDescriptionGerman>Trocken- oder Mörtelmauerwerk</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Grain- or Partikel Size</ColumnDescriptionEnglish>
  </DataColumn>
```

```
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E41 Appellation</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Trocken- oder Mörtelmauerwerk</TableDescriptionGerman>
 <TableDescriptionEnglish>Dry Masonry or Mortar Masonry</TableDescriptionEnglish>
 <OntologyEntity>E55 Type</OntologyEntity>
</Table>

<Table TableName="USBBMaterial" Use="Vocabulary">
  <Column ColumnName="Term" TableName="USBBMaterial">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="010"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <ColumnDescriptionGerman>Material</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Material</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E41 Appellation</OntologyEntity>
 </Column>
  <Column ColumnName="Description" TableName="USBBMaterial">
  <DataColumn>
   <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
   <ColumnDescriptionGerman>Beschreibung</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Description</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P3 has note</OntologyPredicate>
  <OntologyEntity>E62 String</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Material</TableDescriptionGerman>
 <TableDescriptionEnglish>Material</TableDescriptionEnglish>
 <OntologyEntity>E57 Material</OntologyEntity>
</Table>

<Table TableName="USBConsistency" Use="Vocabulary">
  <Column ColumnName="Term" TableName="USBConsistency">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <ColumnDescriptionGerman>Konsistenz</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Consistency</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E41 Appellation</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Konsistenz</TableDescriptionGerman>
 <TableDescriptionEnglish>Consistency</TableDescriptionEnglish>
 <OntologyEntity>E3 Condition State</OntologyEntity>
</Table>

<Table TableName="USBAdditive" Use="Vocabulary">
  <Column ColumnName="Term" TableName="USBAdditive">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <ColumnDescriptionGerman>Zuschlagstoffe</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Additives</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E42 Identifier</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Zuschlagstoffe</TableDescriptionGerman>
 <TableDescriptionEnglish>Additives</TableDescriptionEnglish>
 <OntologyEntity>E57 Material</OntologyEntity>
</Table>

<Table TableName="USBBinding" Use="Storage">
  <Column ColumnName="USNr" TableName="USBBinding">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="06" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="USBuilding" RefColumnName="USNr" DeleteAction="CASCADE"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Kennnummer Stratifikationseinheit</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>ID of the Unit of Stratification</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E42 Identifier</OntologyEntity>
 </Column>
  <Column ColumnName="Humidity" TableName="USBBinding">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="010"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="USBBHumidity" RefColumnName="Term" DeleteAction="RESTRICT"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Trocken- oder Mörtelmauerwerk</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Dry Masonry or Mortar Masonry</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P108 was produced by - E12 Production - P32 used general technique</OntologyPredicate>
  <OntologyEntity>E55 Type</OntologyEntity>
```

```
 </Column>
 <Column ColumnName="Material" TableName="USBBinding">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="010"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="USBBMaterial" RefColumnName="Term" DeleteAction="RESTRICT"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Material</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Material</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P45 consists of</OntologyPredicate>
  <OntologyEntity>E57 Material</OntologyEntity>
 </Column>
 <Column ColumnName="Color" TableName="USBBinding">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="Color" RefColumnName="Code" DeleteAction="RESTRICT" UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Farbe nach Munsell Soil Color Chart</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Color According Munsell Soil Color Chart</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P56 bears feature</OntologyPredicate>
  <OntologyEntity>E25 Man-Made Feature</OntologyEntity>
 </Column>
  <Column ColumnName="GrainSizeMM" TableName="USBBinding">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="USBBGrainSize" RefColumnName="Term" DeleteAction="RESTRICT"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Partikelgröße in Millimeter</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Grain- or Partikel Size in Millimeters</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P39 was measured by - E16 Measurement - P40 observed dimension - E54 Dimension - P90 has value
  </OntologyPredicate>
  <OntologyEntity>E60 Number</OntologyEntity>
 </Column>
  <Column ColumnName="Consistency" TableName="USBBinding">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="USBConsistency" RefColumnName="Term" DeleteAction="RESTRICT"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Konsistenz</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Consistency</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P44 has condition</OntologyPredicate>
  <OntologyEntity>E3 Condition State</OntologyEntity>
 </Column>
  <Column ColumnName="Seam" TableName="USBBinding">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="USBBSeam" RefColumnName="Term" DeleteAction="RESTRICT" UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Fugenbild</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Seam Elaboration</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P56 bears feature</OntologyPredicate>
  <OntologyEntity>E25 Man-Made Feature</OntologyEntity>
 </Column>
  <Column ColumnName="SeamExtents" TableName="USBBinding">
  <DataColumn>
   <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
   <ColumnDescriptionGerman>Fugendimenstionen</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Seam Dimensions</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P39 was measured by</OntologyPredicate>
  <OntologyEntity>E16 Measurement</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Mörtel</TableDescriptionGerman>
 <TableDescriptionEnglish>Binding</TableDescriptionEnglish>
 <OntologyEntity>E22 Man-Made Object</OntologyEntity>
</Table>

<Table TableName="USBBAdditive" Use="Storage">
  <Column ColumnName="USNr" TableName="USBBAdditive">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="06" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="USBBinding" RefColumnName="USNr" DeleteAction="CASCADE"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Kennnummer Stratifikationseinheit</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>ID of the Unit of Stratification</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E42 Identifier</OntologyEntity>
 </Column>
  <Column ColumnName="Additive" TableName="USBBAdditive">
  <DataColumn>
```

```
    <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
    <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
    <Constraint><FOREIGNKEY RefTableName="USBAdditive" RefColumnName="Term" DeleteAction="RESTRICT"
         UpdateAction="CASCADE"/></Constraint>
    <ColumnDescriptionGerman>Zuschlagstoff</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Additive</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P45 consists of</OntologyPredicate>
   <OntologyEntity>E57 Material</OntologyEntity>
  </Column>
   <Column ColumnName="SharePercentage" TableName="USBBAdditive">
   <DataColumn>
    <DataColumnType><NUMERIC Precision="03" Scale="00"/></DataColumnType>
    <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
    <ColumnDescriptionGerman>Anteil des Zuschlagstoffs am Mörtel (in Prozent)</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Share of Additive on Mortal (Percentage)</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P39 was measured by - E16 Measurement - P40 observed dimension - E54 Dimension - P90 has value
   </OntologyPredicate>
   <OntologyEntity>E60 Number</OntologyEntity>
  </Column>
   <Column ColumnName="Notes" TableName="USBBAdditive">
   <DataColumn>
    <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
    <ColumnDescriptionGerman>Anmerkungen</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Notes</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P3 has note</OntologyPredicate>
   <OntologyEntity>E62 String</OntologyEntity>
  </Column>
  <TableDescriptionGerman>Zuschlagstoffe im Mörtel</TableDescriptionGerman>
  <TableDescriptionEnglish>Addtives of the Mortar</TableDescriptionEnglish>
  <OntologyEntity>E19 Physical Object</OntologyEntity>
 </Table>

 <Table TableName="USBPSurfaceTreatment" Use="Vocabulary">
   <Column ColumnName="Term" TableName="USBPSurfaceTreatment">
   <DataColumn>
    <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
    <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
    <ColumnDescriptionGerman>Oberflächenbehandlung</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Surface Treatment</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P1 is identified by</OntologyPredicate>
   <OntologyEntity>E41 Appellation</OntologyEntity>
  </Column>
  <TableDescriptionGerman>Oberflächenbehandlung</TableDescriptionGerman>
  <TableDescriptionEnglish>Surface Treatment</TableDescriptionEnglish>
  <OntologyEntity>E55 Type</OntologyEntity>
 </Table>

 <Table TableName="USBPlaster" Use="Storage">
   <Column ColumnName="USNr" TableName="USBPlaster">
   <DataColumn>
    <DataColumnType><NUMERIC Precision="06" Scale="00"/></DataColumnType>
    <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
    <Constraint><FOREIGNKEY RefTableName="USBuilding" RefColumnName="USNr" DeleteAction="CASCADE"
         UpdateAction="CASCADE"/></Constraint>
    <ColumnDescriptionGerman>Kennnummer Stratifikationseinheit</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>ID of the Unit of Stratification</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P1 is identified by</OntologyPredicate>
   <OntologyEntity>E42 Identifier</OntologyEntity>
  </Column>
   <Column ColumnName="ID" TableName="USBPlaster">
   <DataColumn>
    <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
    <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
    <ColumnDescriptionGerman>Bezeichner des Verputzes</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>ID of the Plaster</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P1 is identified by</OntologyPredicate>
   <OntologyEntity>E42 Identifier</OntologyEntity>
  </Column>
   <Column ColumnName="SurfaceColor" TableName="USBPlaster">
   <DataColumn>
    <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
    <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
    <Constraint><FOREIGNKEY RefTableName="Color" RefColumnName="Code" DeleteAction="RESTRICT" UpdateAction="CASCADE"/></Constraint>
    <ColumnDescriptionGerman>Farbe nach Munsell Soil Color Chart</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Color According Munsell Soil Color Chart</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P56 bears feature</OntologyPredicate>
   <OntologyEntity>E25 Man-Made Feature</OntologyEntity>
  </Column>
   <Column ColumnName="SurfaceTreatment" TableName="USBPlaster">
   <DataColumn>
    <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
    <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
```

```
    <Constraint><FOREIGNKEY RefTableName="USBPSurfaceTreatment" RefColumnName="Term" DeleteAction="RESTRICT"
         UpdateAction="CASCADE"/></Constraint>
    <ColumnDescriptionGerman>Oberflächenbehandlung</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Surface Treatment</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P108 was produced by - E12 Production - P32 used general technique</OntologyPredicate>
   <OntologyEntity>E55 Type</OntologyEntity>
  </Column>
   <Column ColumnName="ThicknessMM" TableName="USBPlaster">
   <DataColumn>
    <DataColumnType><NUMERIC Precision="03" Scale="00"/></DataColumnType>
    <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
    <ColumnDescriptionGerman>Dicke (in Millimeter)</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Thickness (in Millimeters)</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P39 was measured by - E16 Measurement - P40 observed dimension - E54 Dimension - P90 has value
   </OntologyPredicate>
   <OntologyEntity>E60 Number</OntologyEntity>
  </Column>
   <Column ColumnName="Extent" TableName="USBPlaster">
   <DataColumn>
    <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
    <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
    <ColumnDescriptionGerman>Ausdehnung und Verlauf</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Extent and Dispensation</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P39 was measured by - E16 Measurement - P40 observed dimension - E54 Dimension - P90 has value
   </OntologyPredicate>
   <OntologyEntity>E60 Number</OntologyEntity>
  </Column>
   <Column ColumnName="Color" TableName="USBPlaster">
   <DataColumn>
    <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
    <Constraint><FOREIGNKEY RefTableName="Color" RefColumnName="Code" DeleteAction="RESTRICT" UpdateAction="CASCADE"/></Constraint>
    <ColumnDescriptionGerman>Farbe nach Munsell Soil Color Chart</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Color According Munsell Soil Color Chart</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P56 bears feature</OntologyPredicate>
   <OntologyEntity>E25 Man-Made Feature</OntologyEntity>
  </Column>
   <Column ColumnName="Consistency" TableName="USBPlaster">
   <DataColumn>
    <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
    <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
    <Constraint><FOREIGNKEY RefTableName="USBConsistency" RefColumnName="Term" DeleteAction="RESTRICT"
         UpdateAction="CASCADE"/></Constraint>
    <ColumnDescriptionGerman>Konsistenz</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Consistency</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P44 has condition</OntologyPredicate>
   <OntologyEntity>E3 Condition State</OntologyEntity>
  </Column>
   <Column ColumnName="Notes" TableName="USBPlaster">
   <DataColumn>
    <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
    <ColumnDescriptionGerman>Anmerkungen</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Notes</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P3 has note</OntologyPredicate>
   <OntologyEntity>E62 String</OntologyEntity>
  </Column>
  <TableDescriptionGerman>Verputz</TableDescriptionGerman>
  <TableDescriptionEnglish>Plaster</TableDescriptionEnglish>
  <OntologyEntity>E22 Man-Made Object</OntologyEntity>
 </Table>

 <Table TableName="USBPAdditive" Use="Storage">
   <Column ColumnName="USNr" TableName="USBPAdditive">
   <DataColumn>
    <DataColumnType><NUMERIC Precision="06" Scale="00"/></DataColumnType>
    <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
    <Constraint><FOREIGNKEY RefTableName="USBPlaster" RefColumnName="USNr" DeleteAction="CASCADE"
         UpdateAction="CASCADE"/></Constraint>
    <ColumnDescriptionGerman>Kennnummer Stratifikationseinheit</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>ID of the Unit of Stratification</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P1 is identified by</OntologyPredicate>
   <OntologyEntity>E42 Identifier</OntologyEntity>
  </Column>
   <Column ColumnName="Additive" TableName="USBPAdditive">
   <DataColumn>
    <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
    <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
    <Constraint><FOREIGNKEY RefTableName="USBAdditive" RefColumnName="Term" DeleteAction="RESTRICT"
         UpdateAction="CASCADE"/></Constraint>
    <ColumnDescriptionGerman>Zuschlagstoff</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Additive</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P1 is identified by</OntologyPredicate>
```

```
  <OntologyEntity>E42 Identifier</OntologyEntity>
 </Column>
  <Column ColumnName="GrainSizeMM" TableName="USBPAdditive">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="04" Scale="02"/></DataColumnType>
   <ColumnDescriptionGerman>Partikelgröße in Millimeter</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Grain- or Partikel Size in Millimeters</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P39 was measured by - E16 Measurement - P40 observed dimension - E54 Dimension - P90 has value
  </OntologyPredicate>
  <OntologyEntity>E60 Number</OntologyEntity>
 </Column>
  <Column ColumnName="SharePercentage" TableName="USBPAdditive">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="03" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <ColumnDescriptionGerman>Anteil des Zuschlagstoffs am Verputz (in Prozent)</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Share of Additive on Plaster (Percentage)</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P39 was measured by - E16 Measurement - P40 observed dimension - E54 Dimension - P90 has value
  </OntologyPredicate>
  <OntologyEntity>E60 Number</OntologyEntity>
 </Column>
  <Column ColumnName="Notes" TableName="USBPAdditive">
  <DataColumn>
   <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
   <ColumnDescriptionGerman>Anmerkungen</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Notes</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P3 has note</OntologyPredicate>
  <OntologyEntity>E62 String</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Zuschlagstoffe im Mörtel</TableDescriptionGerman>
 <TableDescriptionEnglish>Addtives of the Mortar</TableDescriptionEnglish>
 <OntologyEntity>E19 Physical Object</OntologyEntity>
</Table>

<Table TableName="USHPosition" Use="Vocabulary">
  <Column ColumnName="Term" TableName="USHPosition">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <ColumnDescriptionGerman>Lage der Skelettfunde</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Position of the Bones</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E41 Appellation</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Lage der Skelettfunde</TableDescriptionGerman>
 <TableDescriptionEnglish>Position of the Bones</TableDescriptionEnglish>
 <OntologyEntity>E3 Condition State</OntologyEntity>
</Table>

<Table TableName="USHPreservation" Use="Vocabulary">
  <Column ColumnName="Term" TableName="USHPreservation">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <ColumnDescriptionGerman>Erhaltungszustand</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>State of Preservation</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E41 Appellation</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Erhaltungszustand</TableDescriptionGerman>
 <TableDescriptionEnglish>State of Preservation</TableDescriptionEnglish>
 <OntologyEntity>E3 Condition State</OntologyEntity>
</Table>

<Table TableName="USHDislocation" Use="Vocabulary">
  <Column ColumnName="Term" TableName="USHDislocation">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <ColumnDescriptionGerman>Dislokation</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Dislocation</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E41 Appellation</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Dislokation</TableDescriptionGerman>
 <TableDescriptionEnglish>Dislocation</TableDescriptionEnglish>
 <OntologyEntity>E3 Condition State</OntologyEntity>
</Table>

<Table TableName="USHAge" Use="Vocabulary">
  <Column ColumnName="Term" TableName="USHAge">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="008"/></DataColumnType>
```

```
    <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
    <ColumnDescriptionGerman>Alter eines Menschen</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Age of a Human</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P1 is identified by</OntologyPredicate>
   <OntologyEntity>E41 Appellation</OntologyEntity>
  </Column>
  <TableDescriptionGerman>Alter eines Menschen</TableDescriptionGerman>
  <TableDescriptionEnglish>Age of a Human</TableDescriptionEnglish>
  <OntologyEntity>E3 Condition State</OntologyEntity>
 </Table>

 <Table TableName="USHSex" Use="Vocabulary">
   <Column ColumnName="Term" TableName="USHSex">
   <DataColumn>
    <DataColumnType><NVARCHAR Length="007"/></DataColumnType>
    <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
    <ColumnDescriptionGerman>Geschlecht eines Menschen</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Sex of a Human</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P1 is identified by</OntologyPredicate>
   <OntologyEntity>E41 Appellation</OntologyEntity>
  </Column>
  <TableDescriptionGerman>Geschlecht eines Menschen</TableDescriptionGerman>
  <TableDescriptionEnglish>Sex of a Human</TableDescriptionEnglish>
  <OntologyEntity>E3 Condition State</OntologyEntity>
 </Table>

 <Table TableName="USHUnearthing" Use="Vocabulary">
   <Column ColumnName="Term" TableName="USHUnearthing">
   <DataColumn>
    <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
    <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
    <ColumnDescriptionGerman>Bergungs Technik</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Unearthing Technique</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P1 is identified by</OntologyPredicate>
   <OntologyEntity>E41 Appellation</OntologyEntity>
  </Column>
  <TableDescriptionGerman>Bergungs Technik</TableDescriptionGerman>
  <TableDescriptionEnglish>Unearthing Technique</TableDescriptionEnglish>
  <OntologyEntity>E3 Condition State</OntologyEntity>
 </Table>

 <Table TableName="USHBones" Use="Vocabulary">
   <Column ColumnName="Term" TableName="USHBones">
   <DataColumn>
    <DataColumnType><NVARCHAR Length="030"/></DataColumnType>
    <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
    <ColumnDescriptionGerman>Knochen des menschlichen Körpers</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Bones of the Human Body</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P1 is identified by</OntologyPredicate>
   <OntologyEntity>E41 Appellation</OntologyEntity>
  </Column>
  <TableDescriptionGerman>Knochen des menschlichen Körpers</TableDescriptionGerman>
  <TableDescriptionEnglish>Bones of the Human Body</TableDescriptionEnglish>
  <OntologyEntity>E20 Biological Object</OntologyEntity>
 </Table>

 <Table TableName="USHBurialType" Use="Vocabulary">
   <Column ColumnName="Term" TableName="USHBurialType">
   <DataColumn>
    <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
    <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
    <ColumnDescriptionGerman>Art der Bestattung: Kremation oder Inhumation</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Type of Burial: Cremation or Inhumation</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P1 is identified by</OntologyPredicate>
   <OntologyEntity>E41 Appellation</OntologyEntity>
  </Column>
  <TableDescriptionGerman>Art der Bestattung: Kremation oder Inhumation</TableDescriptionGerman>
  <TableDescriptionEnglish>Type of Burial: Cremation or Inhumation</TableDescriptionEnglish>
  <OntologyEntity>E3 Condtion State</OntologyEntity>
 </Table>

 <Table TableName="USHumanRemains" Use="Storage">
   <Column ColumnName="Individual" TableName="USHumanRemains">
   <DataColumn>
    <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
    <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
    <ColumnDescriptionGerman>Bezeichnung des Individuums</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>ID of the Individual</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P131 is identified by</OntologyPredicate>
   <OntologyEntity>E82 Actor Appellation</OntologyEntity>
  </Column>
   <Column ColumnName="USNr" TableName="USHumanRemains">
   <DataColumn>
```

```
  <DataColumnType><NUMERIC Precision="06" Scale="00"/></DataColumnType>
  <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
  <Constraint><FOREIGNKEY RefTableName="UnitOfStratification" RefColumnName="USNr" DeleteAction="CASCADE"
       UpdateAction="CASCADE"/></Constraint>
  <ColumnDescriptionGerman>Kennnummer Stratifikationseinheit</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>ID of the Unit of Stratification</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P46 forms part of</OntologyPredicate>
 <OntologyEntity>E18 Physical Thing</OntologyEntity>
</Column>
 <Column ColumnName="BurialType" TableName="USHumanRemains">
 <DataColumn>
  <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
  <Constraint><FOREIGNKEY RefTableName="USHBurialType" RefColumnName="Term" DeleteAction="RESTRICT"
       UpdateAction="CASCADE"/></Constraint>
  <ColumnDescriptionGerman>Lage der Menschlichen Überreste</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Position of the Human Remains</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P44 has condition</OntologyPredicate>
 <OntologyEntity>E3 Condition Type</OntologyEntity>
</Column>
 <Column ColumnName="Position" TableName="USHumanRemains">
 <DataColumn>
  <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
  <Constraint><FOREIGNKEY RefTableName="USHPosition" RefColumnName="Term" DeleteAction="RESTRICT"
       UpdateAction="CASCADE"/></Constraint>
  <ColumnDescriptionGerman>Lage der Menschlichen Überreste</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Position of the Human Remains</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P44 has condition</OntologyPredicate>
 <OntologyEntity>E3 Condition Type</OntologyEntity>
</Column>
 <Column ColumnName="Orientation" TableName="USHumanRemains">
 <DataColumn>
  <DataColumnType><NCHAR Length="007"/></DataColumnType>
  <Constraint><FOREIGNKEY RefTableName="Orientation" RefColumnName="ShortTerm" DeleteAction="RESTRICT"
       UpdateAction="CASCADE"/></Constraint>
  <ColumnDescriptionGerman>Kürzel der Ausrichtung</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>ShortTerm of the Direction</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P44 has condition</OntologyPredicate>
 <OntologyEntity>E3 Condition Type</OntologyEntity>
</Column>
 <Column ColumnName="Preservation" TableName="USHumanRemains">
 <DataColumn>
  <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
  <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
  <Constraint><FOREIGNKEY RefTableName="USHPreservation" RefColumnName="Term" DeleteAction="RESTRICT"
       UpdateAction="CASCADE"/></Constraint>
  <ColumnDescriptionGerman>Erhaltungszustand</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>State of Preservation</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P44 has condition</OntologyPredicate>
 <OntologyEntity>E3 Condition Type</OntologyEntity>
</Column>
 <Column ColumnName="Dislocation" TableName="USHumanRemains">
 <DataColumn>
  <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
  <Constraint><FOREIGNKEY RefTableName="USHDislocation" RefColumnName="Term" DeleteAction="RESTRICT"
       UpdateAction="CASCADE"/></Constraint>
  <ColumnDescriptionGerman>Dislokation</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Dislocation</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P44 has condition</OntologyPredicate>
 <OntologyEntity>E3 Condition Type</OntologyEntity>
</Column>
 <Column ColumnName="Age" TableName="USHumanRemains">
 <DataColumn>
  <DataColumnType><NVARCHAR Length="008"/></DataColumnType>
  <Constraint><FOREIGNKEY RefTableName="USHAge" RefColumnName="Term" DeleteAction="RESTRICT" UpdateAction="CASCADE"/></Constraint>
  <ColumnDescriptionGerman>Alter eines Menschen</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Age of a Human</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P44 has condition</OntologyPredicate>
 <OntologyEntity>E3 Condition Type</OntologyEntity>
</Column>
 <Column ColumnName="Sex" TableName="USHumanRemains">
 <DataColumn>
  <DataColumnType><NVARCHAR Length="007"/></DataColumnType>
  <Constraint><FOREIGNKEY RefTableName="USHSex" RefColumnName="Term" DeleteAction="RESTRICT" UpdateAction="CASCADE"/></Constraint>
  <ColumnDescriptionGerman>Geschlecht eines Menschen</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Sex of a Human</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P44 has condition</OntologyPredicate>
 <OntologyEntity>E3 Condition Type</OntologyEntity>
</Column>
 <Column ColumnName="Unearthing" TableName="USHumanRemains">
 <DataColumn>
  <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
```

```
    <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
    <Constraint><FOREIGNKEY RefTableName="USHUnearthing" RefColumnName="Term" DeleteAction="RESTRICT"
         UpdateAction="CASCADE"/></Constraint>
    <ColumnDescriptionGerman>Bergungs Technik</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Unearthing Technique</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P44 has condition</OntologyPredicate>
   <OntologyEntity>E3 Condition Type</OntologyEntity>
  </Column>
   <Column ColumnName="Notes" TableName="USHumanRemains">
   <DataColumn>
    <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
    <ColumnDescriptionGerman>Anmerkungen</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Notes</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P3 has note</OntologyPredicate>
   <OntologyEntity>E62 String</OntologyEntity>
  </Column>
   <Column ColumnName="PrelimDeposDateFrom" TableName="USHumanRemains">
   <DataColumn>
    <DataColumnType><OTHER Type="DATE"/></DataColumnType>
    <ColumnDescriptionGerman>Datierung der Grablegung von (JJJJ-MM-DD)</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Dating of Burial from (JJJJ-MM-DD)</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P25 moved by - E9 Move - P4 has time-span - E52 Time-Span - P79 beginning is qualified by</OntologyPredicate>
   <OntologyEntity>E62 String</OntologyEntity>
  </Column>
   <Column ColumnName="PrelimDeposDateTo" TableName="USHumanRemains">
   <DataColumn>
    <DataColumnType><OTHER Type="DATE"/></DataColumnType>
    <ColumnDescriptionGerman>Datierung der Grablegung bis (JJJJ-MM-DD)</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Dating of Burial to (JJJJ-MM-DD)</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P25 moved by - E9 Move - P4 has time-span - E52 Time-Span - P80 end is qualified by</OntologyPredicate>
   <OntologyEntity>E62 String</OntologyEntity>
  </Column>
  <Column ColumnName="Location" TableName="USHumanRemains">
   <SpatialColumn Dimension="XYZ" Nullable="0">POLYGON</SpatialColumn>
   <OntologyPredicate>P156 occupies - E53 Place - 168 place is defined by</OntologyPredicate>
   <OntologyEntity>E94 Space Primitive</OntologyEntity>
  </Column>
  <Column ColumnName="DrawingSVG" TableName="USHumanRemains">
   <DataColumn>
    <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
    <ColumnDescriptionGerman>SVG Zeichnung</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>SVG Drawing</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P156 occupies - E53 Place - 168 place is defined by</OntologyPredicate>
   <OntologyEntity>E38 Image</OntologyEntity>
  </Column>
  <TableDescriptionGerman>Menschliche Überreste</TableDescriptionGerman>
  <TableDescriptionEnglish>Human Remains</TableDescriptionEnglish>
  <OntologyEntity>E21 Person</OntologyEntity>
 </Table>

 <Table TableName="USHBone" Use="Storage">
   <Column ColumnName="Individual" TableName="USHBone">
   <DataColumn>
    <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
    <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
    <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
    <Constraint><FOREIGNKEY RefTableName="USHumanRemains" RefColumnName="Individual" DeleteAction="CASCADE"
         UpdateAction="CASCADE"/></Constraint>
    <ColumnDescriptionGerman>Bezeichnung des Individuums</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>ID of the Individual</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyEntity>E21 Person</OntologyEntity>
  </Column>
   <Column ColumnName="Term" TableName="USHBone">
   <DataColumn>
    <DataColumnType><NVARCHAR Length="030"/></DataColumnType>
    <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
    <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
    <Constraint><FOREIGNKEY RefTableName="USHBones" RefColumnName="Term" DeleteAction="RESTRICT" UpdateAction="CASCADE"/></Constraint>
    <ColumnDescriptionGerman>Knochen des menschlichen Körpers</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Bones of the Human Body</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyEntity>E20 Biological Object</OntologyEntity>
  </Column>
  <TableDescriptionGerman>Verknüpfungstabelle von Individuum und Knochen</TableDescriptionGerman>
  <TableDescriptionEnglish>Linkage Table of Individual and Bone</TableDescriptionEnglish>
  <OntologyPredicate>P46 is composed of</OntologyPredicate>
 </Table>

 <Table TableName="Find" Use="Storage">
   <Column ColumnName="USNr" TableName="Find">
   <DataColumn>
    <DataColumnType><NUMERIC Precision="06" Scale="00"/></DataColumnType>
    <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
```

```
    <Constraint><FOREIGNKEY RefTableName="UnitOfStratification" RefColumnName="USNr" DeleteAction="RESTRICT"
         UpdateAction="CASCADE"/></Constraint>
    <ColumnDescriptionGerman>Kennnummer Stratifikationseinheit</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>ID of the Unit of Stratification</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P46 forms part of</OntologyPredicate>
   <OntologyEntity>E18 Physical Thing</OntologyEntity>
  </Column>
   <Column ColumnName="ID" TableName="Find">
   <DataColumn>
    <DataColumnType><NVARCHAR Length="030"/></DataColumnType>
    <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
    <ColumnDescriptionGerman>Fund ID</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>ID of the Find</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P1 is identified by</OntologyPredicate>
   <OntologyEntity>E42 Identifier</OntologyEntity>
  </Column>
   <Column ColumnName="DatingFrom" TableName="Find">
   <DataColumn>
    <DataColumnType><OTHER Type="DATE"/></DataColumnType>
    <ColumnDescriptionGerman>Datierung des Fundes von (JJJJ-MM-DD)</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Dating of the Find from (JJJJ-MM-DD)</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P108 was produced by - E12 Production - P4 has time-span - E52 Time-Span - P78 is identified by
   </OntologyPredicate>
   <OntologyEntity>E50 Date</OntologyEntity>
  </Column>
   <Column ColumnName="DatingTo" TableName="Find">
   <DataColumn>
    <DataColumnType><OTHER Type="DATE"/></DataColumnType>
    <ColumnDescriptionGerman>Datierung des Fundes bis (JJJJ-MM-DD)</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Dating of the Find to (JJJJ-MM-DD)</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P93 was taken out of existence by - E64 End of Existence - P4 has time-span - E52 Time-Span - P78 is identified
         by</OntologyPredicate>
   <OntologyEntity>E50 Date</OntologyEntity>
  </Column>
   <Column ColumnName="DrawingSVG" TableName="Find">
   <DataColumn>
    <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
    <ColumnDescriptionGerman>SVG Zeichnung</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>SVG Drawing</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P138 has representation</OntologyPredicate>
   <OntologyEntity>E38 Image</OntologyEntity>
  </Column>
   <Column ColumnName="LinkToModel" TableName="Find">
   <DataColumn>
    <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
    <ColumnDescriptionGerman>Link zu Modell</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Link to Model</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P138 has representation</OntologyPredicate>
   <OntologyEntity>E38 Image</OntologyEntity>
  </Column>
  <Column ColumnName="Location" TableName="Find">
   <SpatialColumn Dimension="XYZ" Nullable="0">POINT</SpatialColumn>
   <OntologyPredicate>P156 occupies - E53 Place - 168 place is defined by</OntologyPredicate>
   <OntologyEntity>E94 Space Primitive</OntologyEntity>
  </Column>
  <TableDescriptionGerman>Funde</TableDescriptionGerman>
  <TableDescriptionEnglish>Finds</TableDescriptionEnglish>
  <OntologyPredicate>E22 Man-Made Object</OntologyPredicate>
 </Table>

 <Table TableName="CerPreservation" Use="Vocabulary">
   <Column ColumnName="Term" TableName="CerPreservation">
   <DataColumn>
    <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
    <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
    <ColumnDescriptionGerman>Erhaltungszustand des Keramikobjekts</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>State of Preservation of the Ceramics Object</ColumnDescriptionEnglish>
   </DataColumn>
   <OntologyPredicate>P1 is identified by</OntologyPredicate>
   <OntologyEntity>E41 Appellation</OntologyEntity>
  </Column>
  <TableDescriptionGerman>Erhaltungszustand des Keramikobjekts</TableDescriptionGerman>
  <TableDescriptionEnglish>State of Preservation of the Ceramics Object</TableDescriptionEnglish>
  <OntologyEntity>E3 Condition State</OntologyEntity>
 </Table>

 <Table TableName="CerVesselPart" Use="Vocabulary">
   <Column ColumnName="Term" TableName="CerVesselPart">
   <DataColumn>
    <DataColumnType><NVARCHAR Length="030"/></DataColumnType>
    <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
    <ColumnDescriptionGerman>Erhaltene Gefäßpartie</ColumnDescriptionGerman>
    <ColumnDescriptionEnglish>Vessel Part</ColumnDescriptionEnglish>
```

```
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E41 Appellation</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Erhaltene Gefäßpartie der Keramik</TableDescriptionGerman>
 <TableDescriptionEnglish>Vessel Part of the Ceramics</TableDescriptionEnglish>
 <OntologyEntity>E22 Man-Made Object</OntologyEntity>
</Table>

<Table TableName="CerSurfaceTreatment" Use="Vocabulary">
  <Column ColumnName="Term" TableName="CerSurfaceTreatment">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <ColumnDescriptionGerman>Oberflächenhehandlung</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Surface Treatment or Elaboration</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E41 Appellation</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Oberflächenhehandlung der Keramik</TableDescriptionGerman>
 <TableDescriptionEnglish>Surface Treatment or Elaboration of the Ceramics</TableDescriptionEnglish>
 <OntologyEntity>E55 Type</OntologyEntity>
</Table>

<Table TableName="CerSurfaceStructure" Use="Vocabulary">
  <Column ColumnName="Term" TableName="CerSurfaceStructure">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <ColumnDescriptionGerman>Oberflächenstruktur</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Surface Structure</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E41 Appellation</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Oberflächenstruktur der Keramik</TableDescriptionGerman>
 <TableDescriptionEnglish>Surface Structure of the Ceramics</TableDescriptionEnglish>
 <OntologyEntity>E25 Man-Made Feature</OntologyEntity>
</Table>

<Table TableName="CerBrilliancy" Use="Vocabulary">
  <Column ColumnName="Term" TableName="CerBrilliancy">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <ColumnDescriptionGerman>Oberflächenglanz</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Brilliancy of the Surface</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E41 Appellation</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Oberflächenglanz</TableDescriptionGerman>
 <TableDescriptionEnglish>Brilliancy of the Surface</TableDescriptionEnglish>
 <OntologyEntity>E3 Condition State</OntologyEntity>
</Table>

<Table TableName="CeramicVessel" Use="Storage">
  <Column ColumnName="ID" TableName="CeramicVessel">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="030"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="Find" RefColumnName="ID" DeleteAction="CASCADE" UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Fund ID</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>ID of the Find</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E42 Identifier</OntologyEntity>
 </Column>
  <Column ColumnName="Preservation" TableName="CeramicVessel">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="CerPreservation" RefColumnName="Term" DeleteAction="RESTRICT"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Erhaltungszustand des Keramikobjekts</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>State of Preservation of the Ceramics Object</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P44 has condition</OntologyPredicate>
  <OntologyEntity>E3 Condition State</OntologyEntity>
 </Column>
  <Column ColumnName="PresSharePercentage" TableName="CeramicVessel">
  <DataColumn>
   <DataColumnType><NUMERIC Precision="03" Scale="00"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <ColumnDescriptionGerman>Erhaltene Menge in Prozent</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Preserved Percentage of the Vessel</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P39 was measured by - E16 Measurement - P40 observed dimension - E54 Dimension - P90 has value
```

```
 </OntologyPredicate>
 <OntologyEntity>E60 Number</OntologyEntity>
</Column>
 <Column ColumnName="RimDiameterMM" TableName="CeramicVessel">
 <DataColumn>
  <DataColumnType><NUMERIC Precision="05" Scale="00"/></DataColumnType>
  <ColumnDescriptionGerman>Randdurchmesser (in Millimeter)</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Rim Diameter (Millimeters)</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P39 was measured by - E16 Measurement - P40 observed dimension - E54 Dimension - P90 has value
 </OntologyPredicate>
 <OntologyEntity>E60 Number</OntologyEntity>
</Column>
 <Column ColumnName="VesselPart" TableName="CeramicVessel">
 <DataColumn>
  <DataColumnType><NVARCHAR Length="030"/></DataColumnType>
  <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
  <Constraint><FOREIGNKEY RefTableName="CerVesselPart" RefColumnName="Term" DeleteAction="RESTRICT"
       UpdateAction="CASCADE"/></Constraint>
  <ColumnDescriptionGerman>Erhaltene Gefäßpartie</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Preserved Vessel Part</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P46 is composed of</OntologyPredicate>
 <OntologyEntity>E22 Man-Made Object</OntologyEntity>
</Column>
 <Column ColumnName="SurfaceTreatmentOut" TableName="CeramicVessel">
 <DataColumn>
  <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
  <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
  <Constraint><FOREIGNKEY RefTableName="CerSurfaceTreatment" RefColumnName="Term" DeleteAction="RESTRICT"
       UpdateAction="CASCADE"/></Constraint>
  <ColumnDescriptionGerman>Oberflächenhehandlung Außenseite</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Surface Treatment or Elaboration on the Outside</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P108 was produced by - E12 Production - P32 used general technique</OntologyPredicate>
 <OntologyEntity>E55 Type</OntologyEntity>
</Column>
 <Column ColumnName="SurfaceTreatmentIn" TableName="CeramicVessel">
 <DataColumn>
  <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
  <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
  <Constraint><FOREIGNKEY RefTableName="CerSurfaceTreatment" RefColumnName="Term" DeleteAction="RESTRICT"
       UpdateAction="CASCADE"/></Constraint>
  <ColumnDescriptionGerman>Oberflächenhehandlung Innenseite</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Surface Treatment or Elaboration on the Inside</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P108 was produced by - E12 Production - P32 used general technique</OntologyPredicate>
 <OntologyEntity>E55 Type</OntologyEntity>
</Column>
 <Column ColumnName="SurfaceTreatmentNotes" TableName="CeramicVessel">
 <DataColumn>
  <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
  <ColumnDescriptionGerman>Anmerkungen zur Oberflächenbehandlung</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Notes about Surface Treatment</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P108 was produced by - E12 Production - P32 used general technique - E55 Type - P3 has note</OntologyPredicate>
 <OntologyEntity>E62 String</OntologyEntity>
</Column>
 <Column ColumnName="SurfaceColorOut" TableName="CeramicVessel">
 <DataColumn>
  <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
  <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
  <Constraint><FOREIGNKEY RefTableName="Color" RefColumnName="Code" DeleteAction="NO ACTION" UpdateAction="CASCADE"/></Constraint>
  <ColumnDescriptionGerman>Farbe nach Munsell Soil Color Chart an der Außenseite</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Color According Munsell Soil Color Chart on the Outside</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P56 bears feature</OntologyPredicate>
 <OntologyEntity>E25 Man-Made Feature</OntologyEntity>
</Column>
 <Column ColumnName="SurfaceColorIn" TableName="CeramicVessel">
 <DataColumn>
  <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
  <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
  <Constraint><FOREIGNKEY RefTableName="Color" RefColumnName="Code" DeleteAction="RESTRICT" UpdateAction="CASCADE"/></Constraint>
  <ColumnDescriptionGerman>Farbe nach Munsell Soil Color Chart an der Innenseite</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Color According Munsell Soil Color Chart on the Inside</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P56 bears feature</OntologyPredicate>
 <OntologyEntity>E25 Man-Made Feature</OntologyEntity>
</Column>
 <Column ColumnName="SurfaceStructure" TableName="CeramicVessel">
 <DataColumn>
  <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
  <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
  <Constraint><FOREIGNKEY RefTableName="CerSurfaceStructure" RefColumnName="Term" DeleteAction="RESTRICT"
       UpdateAction="CASCADE"/></Constraint>
  <ColumnDescriptionGerman>Oberflächenstruktur</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Surface Structure</ColumnDescriptionEnglish>
 </DataColumn>
```

```
 <OntologyPredicate>P56 bears feature</OntologyPredicate>
 <OntologyEntity>E25 Man-Made Feature</OntologyEntity>
</Column>
 <Column ColumnName="Brilliancy" TableName="CeramicVessel">
 <DataColumn>
  <DataColumnType><NVARCHAR Length="025"/></DataColumnType>
  <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
  <Constraint><FOREIGNKEY RefTableName="CerBrilliancy" RefColumnName="Term" DeleteAction="RESTRICT"
       UpdateAction="CASCADE"/></Constraint>
  <ColumnDescriptionGerman>Oberflächenglanz</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Brilliancy of the Surface</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P44 has condition</OntologyPredicate>
 <OntologyEntity>E3 Condition State</OntologyEntity>
</Column>
 <Column ColumnName="Decoration" TableName="CeramicVessel">
 <DataColumn>
  <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
  <ColumnDescriptionGerman>Anmerkungen zu Dekor</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Notes about Decoration</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P56 bears feature</OntologyPredicate>
 <OntologyEntity>E25 Man-Made Feature</OntologyEntity>
</Column>
 <Column ColumnName="FractColorCore" TableName="CeramicVessel">
 <DataColumn>
  <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
  <Constraint><FOREIGNKEY RefTableName="Color" RefColumnName="Code" DeleteAction="RESTRICT" UpdateAction="CASCADE"/></Constraint>
  <ColumnDescriptionGerman>Farbe nach Munsell Soil Color Chart im Zentrum des Bruchs</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Color According Munsell Soil Color Chart on the Center of the Fraction</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P56 bears feature</OntologyPredicate>
 <OntologyEntity>E25 Man-Made Feature</OntologyEntity>
</Column>
 <Column ColumnName="FractColMantelOut" TableName="CeramicVessel">
 <DataColumn>
  <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
  <Constraint><FOREIGNKEY RefTableName="Color" RefColumnName="Code" DeleteAction="RESTRICT" UpdateAction="CASCADE"/></Constraint>
  <ColumnDescriptionGerman>Farbe nach Munsell Soil Color Chart im äußeren Mantel des Bruchs</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Color According Munsell Soil Color Chart on the outer Mantel of the Fraction</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P56 bears feature</OntologyPredicate>
 <OntologyEntity>E25 Man-Made Feature</OntologyEntity>
</Column>
 <Column ColumnName="FractColMantelIn" TableName="CeramicVessel">
 <DataColumn>
  <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
  <Constraint><FOREIGNKEY RefTableName="Color" RefColumnName="Code" DeleteAction="RESTRICT" UpdateAction="CASCADE"/></Constraint>
  <ColumnDescriptionGerman>Farbe nach Munsell Soil Color Chart im inneren Mantel des Bruchs</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Color According Munsell Soil Color Chart on the inner Mantel of the Fraction</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P56 bears feature</OntologyPredicate>
 <OntologyEntity>E25 Man-Made Feature</OntologyEntity>
</Column>
 <Column ColumnName="FractColSurfaceOut" TableName="CeramicVessel">
 <DataColumn>
  <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
  <Constraint><FOREIGNKEY RefTableName="Color" RefColumnName="Code" DeleteAction="RESTRICT" UpdateAction="CASCADE"/></Constraint>
  <ColumnDescriptionGerman>Farbe nach Munsell Soil Color Chart am äußeren Rand des Bruchs</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Color According Munsell Soil Color Chart on the outer Edge of the Fraction</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P56 bears feature</OntologyPredicate>
 <OntologyEntity>E25 Man-Made Feature</OntologyEntity>
</Column>
 <Column ColumnName="FractColSurfaceIn" TableName="CeramicVessel">
 <DataColumn>
  <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
  <Constraint><FOREIGNKEY RefTableName="Color" RefColumnName="Code" DeleteAction="RESTRICT" UpdateAction="CASCADE"/></Constraint>
  <ColumnDescriptionGerman>Farbe nach Munsell Soil Color Chart am inneren Rand des Bruchs</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Color According Munsell Soil Color Chart on the inner Edge of the Fraction</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P56 bears feature</OntologyPredicate>
 <OntologyEntity>E25 Man-Made Feature</OntologyEntity>
</Column>
 <Column ColumnName="Product" TableName="CeramicVessel">
 <DataColumn>
  <DataColumnType><NVARCHAR Length="025"/></DataColumnType>
  <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
  <ColumnDescriptionGerman>Ware</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Product</ColumnDescriptionEnglish>
 </DataColumn>
 <OntologyPredicate>P2 has type - E55 Type - P2 has type - E55 Type - P2 has type</OntologyPredicate>
 <OntologyEntity>E55 Type</OntologyEntity>
</Column>
 <Column ColumnName="Type" TableName="CeramicVessel">
 <DataColumn>
  <DataColumnType><NVARCHAR Length="025"/></DataColumnType>
  <ColumnDescriptionGerman>Typ</ColumnDescriptionGerman>
  <ColumnDescriptionEnglish>Type</ColumnDescriptionEnglish>
```

```
  </DataColumn>
  <OntologyPredicate>P2 has type - E55 Type - P2 has type</OntologyPredicate>
  <OntologyEntity>E55 Type</OntologyEntity>
 </Column>
  <Column ColumnName="Variant" TableName="CeramicVessel">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="025"/></DataColumnType>
   <ColumnDescriptionGerman>Variante</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Variant</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P2 has type</OntologyPredicate>
  <OntologyEntity>E55 Type</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Keramikgefäße</TableDescriptionGerman>
 <TableDescriptionEnglish>Ceramic Vessels</TableDescriptionEnglish>
 <OntologyEntity>E22 Man-Made Object</OntologyEntity>
</Table>

<Table TableName="Coat" Use="Storage">
  <Column ColumnName="VesselID" TableName="Coat">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="030"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="CeramicVessel" RefColumnName="ID" DeleteAction="CASCADE"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Fund ID</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>ID of the Find</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P46 forms part of</OntologyPredicate>
  <OntologyEntity>E22 Man-Made Object</OntologyEntity>
 </Column>
  <Column ColumnName="CoatID" TableName="Coat">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="020"/></DataColumnType>
   <Constraint><SimpleConstraint Type="PRIMARY KEY"/></Constraint>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <ColumnDescriptionGerman>Überzug ID</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>ID of the Coat</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E42 Identifier</OntologyEntity>
 </Column>
  <Column ColumnName="Brilliancy" TableName="Coat">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="025"/></DataColumnType>
   <Constraint><SimpleConstraint Type="NOT NULL"/></Constraint>
   <Constraint><FOREIGNKEY RefTableName="CerBrilliancy" RefColumnName="Term" DeleteAction="RESTRICT"
        UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Oberflächenglanz</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Brilliancy of the Surface</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P1 is identified by</OntologyPredicate>
  <OntologyEntity>E3 Composition State</OntologyEntity>
 </Column>
  <Column ColumnName="Decoration" TableName="Coat">
  <DataColumn>
   <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
   <ColumnDescriptionGerman>Anmerkungen zu Dekor</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Notes about Decoration</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P56 bears feature</OntologyPredicate>
  <OntologyEntity>E25 Man-Made Feature</OntologyEntity>
 </Column>
  <Column ColumnName="Extent" TableName="Coat">
  <DataColumn>
   <DataColumnType><OTHER Type="TEXT"/></DataColumnType>
   <ColumnDescriptionGerman>Lage und Ausdehnung des Überzugs</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Position and Extent of the Coat</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P39 was measured by</OntologyPredicate>
  <OntologyEntity>E16 Measurement</OntologyEntity>
 </Column>
  <Column ColumnName="Color" TableName="Coat">
  <DataColumn>
   <DataColumnType><NVARCHAR Length="015"/></DataColumnType>
   <Constraint><FOREIGNKEY RefTableName="Color" RefColumnName="Code" DeleteAction="RESTRICT" UpdateAction="CASCADE"/></Constraint>
   <ColumnDescriptionGerman>Farbe nach Munsell Soil Color Chart</ColumnDescriptionGerman>
   <ColumnDescriptionEnglish>Color According Munsell Soil Color Chart</ColumnDescriptionEnglish>
  </DataColumn>
  <OntologyPredicate>P56 bears feature</OntologyPredicate>
  <OntologyEntity>E25 Man-Made Feature</OntologyEntity>
 </Column>
 <TableDescriptionGerman>Überzug</TableDescriptionGerman>
 <TableDescriptionEnglish>Coat</TableDescriptionEnglish>
 <OntologyEntity>E22 Man-Made Object</OntologyEntity>
</Table>
</Database>
```

[DATASTRUCTURE].PHP

```php
<?php
//*****************************************************************************
//Written by Thomas Winklehner
//on 25.10.2016
//Purpose:
//        HTML/PHP-Webpage
//*****************************************************************************

//*****************************************************************************
//                          Get and Display Datastructure
//*****************************************************************************

    $ressourceDir = realpath(__DIR__);
    for($count = 0; $count < 3; $count++){ $ressourceDir = dirname($ressourceDir); };

    $currentDatastructName = basename(__file__, '.php');
    $scriptsDir = $ressourceDir . DIRECTORY_SEPARATOR . 'Scripts' . DIRECTORY_SEPARATOR;

    function displayDatastruct(){
        global $ressourceDir;
        global $currentDatastructName;
        global $scriptsDir;

        if($ressourceDir !== '.'){
            $datastructHTMLDir = $ressourceDir
                . DIRECTORY_SEPARATOR
                . 'DatabaseStructuresHTML'
                . DIRECTORY_SEPARATOR;

            $htmlDataStruc = $datastructHTMLDir . $currentDatastructName . '.html';
            if(file_exists($htmlDataStruc)){
                $content = file_get_contents($htmlDataStruc);
                echo $content === false? '' : $content;
            }
            else{
                $htmlDataStructGenerator = $scriptsDir . 'GetDatastructFromXML.php';
                if(file_exists($htmlDataStructGenerator)){
                    include_once $htmlDataStructGenerator;
                    $returnValue = createDatastructHTMLFile($currentDatastructName);
                    if($returnValue === true){
                        $content = file_get_contents($htmlDataStruc);
                        echo $content === false? '' : $content;
                    }
                }
            };
        };
    };
    $GetRepositoriesScript = $scriptsDir . "MainConnectionUtility.php";
    include_once $GetRepositoriesScript;
?>

<html>
    <head>
        <meta charset="utf-8"/>
        <link href="ScriptsAndStyle/css.css" rel="stylesheet" title="Default Style"/>
        <script src="ScriptsAndStyle/JQueryCompressed.js" type="text/javascript"></script>
        <script src="ScriptsAndStyle/cytoscape.js" type="text/javascript"></script>
        <script src="ScriptsAndStyle/MainJSScript.js" type="text/javascript"></script>
        <script src="ScriptsAndStyle/openlayers.js" type="text/javascript"></script>
        <script src="ScriptsAndStyle/jscolor.js" type="text/javascript"></script>
        <script src="ScriptsAndStyle/three.js" type="text/javascript"></script>
        <script src="ScriptsAndStyle/TrackballControls.js" type="text/javascript"></script>
        <script src="ScriptsAndStyle/GeometryUtils.js" type="text/javascript"></script>
        <title></title>
    </head>

    <body>

        <!-- ******************************Fixed-Content****************************** -->

        <div id="fixed_content"
            <!-- Main Navigation Menu -->
            <nav id="main_menu">
                <ul>
                    <li><a href='#datastruct_diagram'>Datastructure</a></li>
                    <li><a href='#query_area'>Query</a></li>
                    <li><a href='#query_results_area'>QueryResults</a></li>
                    <li><a href='#ThreeD_map_area'>3D</a></li>
                    <li><a href='#map_area'>Mapping</a></li>
                    <li><a href='#wms_wfs_area'>WMS/WFS</a></li>
                </ul>
            </nav>

            <!-- Datastructure: Tables, Columns etc. -->
            <div id="datastruct"><?php displayDatastruct(); ?></div>
        </div>
```

```
<!-- ****************************Main-Content***************************** -->

<!-- 2D Map, QueryArea etc. -->
<div id="main_content">

    <!-- ************************Datastructure-Diagram************************ -->

    <!-- Datastruct-Diagram -->
    <div id="datastruct_diagram">
        <h1><?php echo $currentDatastructName . ' - Structure Diagram'; ?></h1>
        <div id="cy"></div>
        <div id="datastruct_panel">
            <form action="" class="datastruct_panel" id="datastruct_panel01">
                <p>Tables</p>
                <select name="datastruct_panel_tables"></select>
                <p>Columns</p>
                <select name="datastruct_panel_columns"></select>

            </form>
            <form action="" class="datastruct_panel" id="datastruct_panel02">
                <p>Diagram-Type</p>
                <button type="button" value="circle">circle</button><br>
                <button type="button" value="concentric">concentric</button><br>
                <button type="button" value="breadthfirst">tree(single root)</button><br>
                <button type="button" value="breadthfirst_roots">tree(mult. roots)</button><br>
            </form>
            <form action="" class="datastruct_panel" id="datastruct_panel03">
                <p>Size-Controls</p>
                <select name="datastruct_panel_linewidth"></select><label> px Line</label>
                <select name="datastruct_panel_fontheight"></select><label> px Font</label>
            </form>
            <form action="" class="datastruct_panel" id="datastruct_panel04">
                <p>Downloads</p>
                <button type='button' id='download_datastructure_schema'>Datastruct-Schema</button>
                <button type='button' id='download_datastructure'>Datastructure</button>
                <button type='button' id='download_glossary_schema'>Glossary-Schema</button>
            </form>
        </div>
    </div>

    <!-- ************************Query-Composer************************ -->

    <!-- Query-Composer -->
    <div id="query_area">
        <h1>Query</h1>
        <form action="" id="query_form">
            <!-- Repositorieslist -->
            <div id="repositories">
                <div id="repositories_container">
                    <h2>Repositories</h2>
                    <!-- from MainConnectionUtility.php : -->
                    <ul id="repositories_list"><?php echo getRepositoriesHTML($currentDatastructName); ?></ul>
                </div>

                <div id="limit_offset_send_container">
                    <p>limit</p>
                    <input name="query_limit" type='number' min='0' max='10000' value='100'/><br>
                    <p>offset</p>
                    <input name="query_offset" type='number' min='0' value='0'/><br>
                    <button type="button" value="query_send">Send Query</button>
                </div>

                <div id="binary_download">
                    <p>Download</p>
                    <input type='text' value='Filename.suffix'/><br>
                    <button type="button">get Repository</button>
                </div>

            </div>

            <!-- Composer -->
            <div id="sql_composer">
                <!-- From-Tables -->
                <div id="from_tables_container">
                    <h2>Tables</h2>
                    <ul id="from_tables_list"></ul>
                    <button type="button" value="setFirstFromTable">Set First Table</button><br>
                    <button type="button" value="clear">Clear</button>

                    <!-- Order By List -->
                    <div id="orderBy_container">
                        <h2>Order by</h2>
                        <ul id="orderBy_list"></ul>
                    </div>
                </div>
                <div id="selected_columns_container">
                    <h2>Columns</h2>
                    <ul id="selected_columns_list"></ul>
                </div>
```

```
        </div>
    </form>
</div>

<!-- ************************Query-Results-Table************************ -->

<!-- Query-Results -->
<div id="query_results_area">
    <h1>Query Results</h1>
    <div id="query_results_container"></div>
    <button type="button">Add To Map</button>
    <button type="button">View in 3D</button>
</div>

<!-- ************************3D-Visualisiation-Area************************ -->

<!-- Query-Results -->
<div id="ThreeD_map_area">
    <h1>3D Mapping</h1>
    <!-- Layers-Controls -->
    <div id=ThreeD_map_layers_manager>
        <input type='text' name='ThreeD_map_color' class='jscolor'/><span>color</span>
        <select></select>
        <button type="button">Zoom To</button>
        <button type="button">Reset All</button>
        <button type="button">></button>
        <button type="button"><</button>
    </div>
    <div id="ThreeD_map_geometrieslist_container">
        <h3>Features</h3>
        <ul id='ThreeD_map_geometrieslist'></ul>
    </div>

    <div id="ThreeD_map_visualisation_area"></div>
</div>

<!-- ************************Map-Area************************ -->

<div id="map_area">
    <h1>Mapping</h1>

    <!-- Layers-Controls -->
    <div id=map_layers_manager>
        <button type="button" name='map_setStyle_button'>Set</button>
        <input type='text' name='current_vectorlayer_stroke' class='jscolor'/><span>stroke</span>
        <input type='text' name='current_vectorlayer_fill' class='jscolor'/><span>fill</span>
        <select name="current_vectorlayer"></select></select><span>current:</span>
        <button type="button" name='map_reset_button'>Reset Map</button>
        <button type="button" name='map_new_vectorlayer_button'>New Layer</button>
        <input type='text' name='map_new_vectorlayer_name'/><span>Layername:</span>
    </div>

    <!-- Layers -->
    <div id="map_layers_area">
        <h3>Basemaps</h3>
        <ul id='map_layers_raster_list'>
            <li>
                <input type="checkbox" id="map_basmaps_visibility_0" checked/>
                <label>OpenStreetMap</label>
            </li>
        </ul>

        <h3>Vectorlayers</h3>
        <ul id='map_layers_vector_list'></ul>
    </div>

    <!-- Map -->
    <div id="map"></div>
</div>

<!-- WMS-WFS -->
<div id="wms_wfs_area">
    <div id="wms_area">
        <h1>load WMS (tiled)</h1>
        <h4>Version 1.3.0</h4>
        <h4>EPSG:3857</h4>
        <p>URL:</p>
        <input id="wms_url" type=text title="url"/>
        <p>Layer:</p>
        <input type=text id="wms_layer" title="Layer-Name"/>
        <button type='button'>Request</button>
    </div>

    <div id="wfs_area">
        <h1>load WFS</h1>
        <h4>EPSG:3857</h4>
        <p>URL:</p>
        <input id="wfs_url" type=text title="url"/>
        <p>typename</p>
```

```
                    <input type=text id="wfs_layer" title="Layer-Name"/>
                    <p>output-format</p>
                    <select>
                        <option value='json'>json</option>
                        <option value='application/json'>application/json</option>
                        <option value='gml'>gml</option>
                        <option value='GML'>GML</option>
                        <option value='gml3'>gml3</option>
                        <option value='GML3'>GML3</option>
                        <option value='kml'>kml</option>
                        <option value='KML'>KML</option>
                    </select><br/>
                    <button type='button'>Request</button>
                </div>
            </div>

        </div>

    </body>
</html>
```

COLUMN.PHP

```php
<?php

//*****************************************************************************
//Written by Thomas Winklehner
//on 22.10.2016
//Purpose:
//       - provide Columns-Classes for Database-Structure emulation
//       - helping outputting Database-Structure as valid HTML from XML
//*****************************************************************************


//*****************************************************************************
//                                  INTERFACE
//*****************************************************************************

    interface ColumnsInterface{
        function isDataColumn();
        function isSpatialColumn();
        function ToHTML();
        function setTable($TableName);
    }

//*****************************************************************************
//                             BASIC COLUMN CLASS
//*****************************************************************************

    class DataColumn implements ColumnsInterface{

        public function __construct(
            $ColumnName,
            $ColumnType,
            $ColumnDescGerm,
            $ColumnDescEngl){
                $this->Name = (string)$ColumnName;
                $this->Type = (string)$ColumnType;

                if($this->Type === 'REAL' || $this->Type === 'INTEGER'){ $this->Category = 'NUMBER'; }
                else{ $this->Category = $this->Type; }

                $this->DescGerm = (string)$ColumnDescGerm;
                $this->DescEngl = (string)$ColumnDescEngl;
                $this->table = "";
        }

        function isDataColumn(){ return true; }

        function isSpatialColumn(){ return false; }

        public function getName(){ return $this->Name; }

        public function getType(){ return $this->Type; }

        public function getColumnDescriptionGerman(){ return $this->DescGerm; }

        public function getColumnDescriptionEnglish(){ return $this->DescEngl; }

        public function setPrimaryKey() { $this->primaryKey = true; }

        public function setForeignKey($table, $column) {
            if($this->Type != "BLOB" && $this->Type != "TEXT" &&
                is_string($table) && is_string($column)){
                    $this->refTable = $table;
                    $this->refColumn = $column;
            }
        }

        public function isPrimaryKey() {
            if(isset($this->primaryKey) && $this->primaryKey === true) { return true; }
            else{ return false; }
        }

        public function isForeignKey() {
            if(isset($this->refTable) && isset($this->refColumn)) { return true; }
            else{ return false; }
        }

        public function setTable($tableName){
            if(isset($tableName) && is_string($tableName)){
                $this->table = (string)$tableName;
            };
        }

        public function getForeignKeyTable() {
            if(isset($this->refTable)) { return $this->refTable; }
            else{ return ""; }
        }
```

```
        public function getForeignKeyColumn() {
            if(isset($this->refColumn)) { return $this->refColumn; }
            else{ return ""; }
        }

        function ToHTML(){
            $text =
                "\t<li class=\"datastruct_column\" id=\"column_{$this->Name}_{$this->table}\">\n"
                    . "\t\t<p class=\"datastruct_column_name\">" . htmlentities($this->Name) . "</p>\n"
                    . "\t\t<p class=\"datastruct_column_cat\">" . htmlentities($this->Category) . "</p>\n"
                    . "\t\t<p class=\"datastruct_column_type\">" . htmlentities($this->Type) . "</p>\n"
                    . "\t\t<p class=\"datastruct_column_desc\">" . htmlentities($this->DescEngl) . "</p>\n"
                    . "\t\t<p class=\"datastruct_column_desc\">" . htmlentities($this->DescGerm) . "</p>\n";
            if(isset($this->refTable) && isset($this->refColumn)) {
                $text
                    .= "\t\t<p class=\"datastruct_FK datastruct_tableFK\">FK-Table: "
                        . "<span href=\"#table_$this->refTable\">" . htmlentities($this->refTable) .
                   "</span>"
                    ."</p>\n"

                    . "\t\t<p class=\"datastruct_FK datastruct_columnFK\">FK-Column: "
                        . "<span href=\"#column_" . $this->refColumn . "_" + $this->refTable . "\">" .
                   htmlentities($this->refColumn) . "</span>"
                    ."</p>\n";
            }
            $text .= "\t</li>\n";
            return $text;
        }

        protected $Name;
        protected $Type;
        protected $Category;
        protected $DescGerm;
        protected $DescEngl;
        protected $table;
    }

//*****************************************************************************
//                            CHARACTER COLUMN CLASS
//*****************************************************************************

    class CharacterColumn extends DataColumn{

        public function __construct(
            $ColumnName,
            $ColumnType,
            $ColumnDescGerm,
            $ColumnDescEngl,
            $length){
                parent::__construct(
                    $ColumnName,
                    $ColumnType,
                    $ColumnDescGerm,
                    $ColumnDescEngl);
                $this->length = (integer)$length;
        }

        public function getLength(){ return $this->length; }

        public function setTable($tableName){ parent::setTable($tableName); }

        function ToHTML(){
            $text =
                "\t<li class=\"datastruct_column\" id=\"column_{$this->Name}_{$this->table}\">\n"
                    . "\t\t<p class=\"datastruct_column_name\">"  . htmlentities($this->Name) . "</p>\n"
                    . "\t\t<p class=\"datastruct_column_cat\">TEXT</p>\n"
                    . "\t\t<p class=\"datastruct_column_type\">" . htmlentities($this->Type . '(' . $this->length) . ")</p>\n"
                    . "\t\t<p class=\"datastruct_column_desc\">" . htmlentities($this->DescEngl) . "</p>\n"
                    . "\t\t<p class=\"datastruct_column_desc\">" . htmlentities($this->DescGerm) . "</p>\n";
            if(isset($this->refTable) && isset($this->refColumn)) {
                $text .= "\t\t<p class=\"datastruct_FK datastruct_tableFK\">FK-Table: "
                        . "<span href=\"#table_$this->refTable\">" . htmlentities($this->refTable) . "</span>"
                    ."</p>\n"

                    . "\t\t<p class=\"datastruct_FK datastruct_columnFK\">FK-Column: "
                        . "<span href=\"#column_" . $this->refColumn . "_" . $this->refTable . "\">" . htmlentities($this->refColumn) .
                           "</span>"
                    ."</p>\n";
            }
            $text .= "\t</li>\n";
            return $text;
        }

        private $length;
    }

//*****************************************************************************
//                            NUMERIC COLUMN CLASS
```

```
//*****************************************************************************

    class NumericColumn extends DataColumn{

        public function __construct(
            $ColumnName,
            $ColumnType,
            $ColumnDescGerm,
            $ColumnDescEngl,
            $precision,
            $scale){
                parent::__construct(
                    $ColumnName,
                    $ColumnType,
                    $ColumnDescGerm,
                    $ColumnDescEngl);
                $this->precision = (integer)$precision;
                $this->scale = (integer)$scale;
        }

        public function getPrecision(){ return $this->precision; }

        public function getScale(){ return $this->scale; }

        public function setTable($tableName){ parent::setTable($tableName); }

        function ToHTML(){
            $text =
                "\t<li class=\"datastruct_column\" id=\"column_{$this->Name}_{$this->table}\">\n"
                    . "\t\t<p class=\"datastruct_column_name\">"   . htmlentities($this->Name) . "</p>\n"
                    . "\t\t<p class=\"datastruct_column_cat\">NUMBER</p>\n"
                    . "\t\t<p class=\"datastruct_column_type\">" . htmlentities($this->Type . '(' . $this->precision . '/' . $this-
                      >scale) . ")</p>\n"
                    . "\t\t<p class=\"datastruct_column_desc\">" . htmlentities($this->DescEngl) . "</p>\n"
                    . "\t\t<p class=\"datastruct_column_desc\">" . htmlentities($this->DescGerm) . "</p>\n";
            if(isset($this->refTable) && isset($this->refColumn)) {
                $text .= "\t\t<p class=\"datastruct_FK datastruct_tableFK\">FK-Table: "
                        . "<span href=\"#table_$this->refTable\">" . htmlentities($this->refTable) . "</span>"
                    ."</p>\n"

                    . "\t\t<p class=\"datastruct_FK datastruct_columnFK\">FK-Column: "
                        . "<span href=\"#column_" . $this->refColumn . "_" . $this->refTable . "\">" . htmlentities($this->refColumn) .
                      "</span>"
                    ."</p>\n";
            }
            $text .= "\t</li>\n";
            return $text;
        }

        private $precision;
        private $scale;
    }

//*****************************************************************************
//                            SPATIAL COLUMN CLASS
//*****************************************************************************

    class SpatialColumn implements ColumnsInterface{

        public function __construct(
            $ColumnName,
            $ColumnType,
            $dimension){
                $this->Name = (string)$ColumnName;
                $this->Type = (string)$ColumnType;
                $this->dimension = (string)$dimension;
                $this->table = "";
        }

        function isDataColumn(){ return false; }

        function isSpatialColumn(){ return true; }

        public function getName(){ return $this->Name; }

        public function getType(){ return $this->Type; }

        public function getDimension(){ return $this->dimension; }

        public function setTable($tableName){
            if(isset($tableName) && is_string($tableName)){
                $this->table = (string)$tableName;
            };
        }

        function ToHTML(){
            $text =
                "\t<li class=\"datastruct_column\" id=\"column_{$this->Name}_{$this->table}\">\n"
                    . "\t\t<p class=\"datastruct_column_name\">"   . htmlentities($this->Name) . "</p>\n"
```

```
                . "\t\t<p class=\"datastruct_column_cat\">GEOMETRY</p>\n"
                . "\t\t<p class=\"datastruct_column_type\">"   . htmlentities($this->Type) . "</p>\n"
                . "\t\t<p class=\"datastruct_column_dimension\">" . htmlentities($this->dimension) . "</p>\n"
            . "\t</li>\n";
        return $text;
    }

    private $Name;
    private $Type;
    private $dimension;
    private $table;
}

?>
```

TABLE.PHP

```php
<?php

//*****************************************************************************
//Written by Thomas Winklehner
//on 22.10.2016
//Purpose:
//       - provide Table-Class for Database-Structure emulation
//       - helping outputting Database-Structure as valid HTML from XML
//*****************************************************************************

//*****************************************************************************
//                            TABLE CLASS
//*****************************************************************************

    class Table{

        public function __construct(
            $TableName,
            $TableType,
            $TableDescGerm,
            $TableDescEngl){
                $this->Name = (string)$TableName;
                $this->Type = (string)$TableType;
                $this->DescGerm = (string)$TableDescGerm;
                $this->DescEngl = (string)$TableDescEngl;
                $this->Columns = array();
                $this->ReferencedTables = array();
                $this->Rank = -1;
        }

        public function getName(){
            return $this->Name;
        }

        public function getType(){
            return $this->Type;
        }

        public function setRank($newRank){
            if(is_integer($newRank)){ $this->Rank = $newRank; };
        }

        public function rankIsSet(){
            if($this->Rank < 0){ return false; }
            else { return true; };
        }

        public function getRank(){
            return $this->Rank;
        }

        public function getTableDescriptionGerman(){
            return $this->DescGerm;
        }

        public function getTableDescriptionEnglish(){
            return $this->DescEngl;
        }

        public function getColumn($columnName){
            if(is_string($columnName) && array_key_exists($columnName, $this->Columns)){
                return $this->Columns[$columnName];
            }else { return null; };
        }

        public function getColumns(){ return $this->Columns; }

        public function addColumn(ColumnsInterface &$column){
            if(!in_array($column, $this->Columns) &&
               !array_key_exists($column->getName(), $this->Columns)){
                    $column->setTable($this->Name);
                    $this->Columns[$column->getName()] = $column;
                    if($column->isDataColumn() && $column->isForeignKey()){
                        $this->ReferencedTables[] = $column->getForeignKeyTable();
                    };
            }
        }

        public function referencesTables(){
            if(sizeof($this->ReferencedTables) > 0){ return true; }
            else{ return false; };
        }

        public function referencesTable(&$tableName){
            if(is_string($tableName) && in_array($tableName, $this->ReferencedTables)){ return true; }
            else { return false; };
```

```
        }

        public function getReferencedTables(){
            return $this->ReferencedTables;
        }

        function ToHTML(){
            $text =
                "<li class=\"datastruct_table\" id=\"table_{$this->Name}\">\n"
                    ."\t<p>" . htmlentities($this->Name) . "</p>\n";
            if($this->Rank === 0){
                $text .= "\t<p class=\"table_is_glossary\">Glossary</p>\n";
                $text .= "\t<button type='button' value='{$this->Name}'>get xml</button>\n";
            };

            $text .= "\t<p class=\"table_rank\">" . $this->Rank . "</p>\n"
                    ."\t<p class=\"datastruct_table_desc\">" . htmlentities($this->DescGerm) . "</p>\n"
                    ."\t<p class=\"datastruct_table_desc\">" . htmlentities($this->DescEngl) . "</p>\n"
                    ."\t<ul class=\"datastruct_columns_list\" id=\"table_{$this->Name}_columns\">\n"
                    ."\t<p class=\"columns_tag\">Columns:</p>\n";

            foreach ($this->Columns as $Column) {
                $text .= $Column->toHTML();
            }

            $text .=  "\t</ul>\n</li>\n";
            return $text;
        }

        private $Name;
        private $Type;
        private $DescGerm;
        private $DescEngl;
        private $Columns;
        private $ReferencedTables;
        private $Rank;
    }

?>
```

GETDATASTRUCTFROMXML.PHP

```php
<?php
//*****************************************************************************
//Written by Thomas Winklehner
//on 22.10.2016
//*****************************************************************************

//*****************************************************************************
//                              Globals
//*****************************************************************************

    define('ScriptsDirectory', __DIR__ . DIRECTORY_SEPARATOR);
    define('HomeDirectory', dirname(realpath(__DIR__)) . DIRECTORY_SEPARATOR);
    define('DatastructsDirectory', HomeDirectory . 'DatabaseStructures' . DIRECTORY_SEPARATOR);
    define('DatastructsHTMLDirectory', HomeDirectory . 'DatabaseStructuresHTML' . DIRECTORY_SEPARATOR);
     define('DatastructsSchema', HomeDirectory . 'XMLDataBaseStructure' . DIRECTORY_SEPARATOR . '   DatabaseStructureSchema.xsd');

//*****************************************************************************
//                            Function: reading XML
//*****************************************************************************

    function generateTablesFromXMLDatastructure($datastructName){

        //check Existence of Files
        $DatastructFilePath = DatastructsDirectory . $datastructName . '.xml';
        if(!file_exists($DatastructFilePath) || !file_exists(DatastructsSchema)){ return false; };

        //validate XML
        $xmlDoc = new DOMDocument();
        $xmlDoc->load($DatastructFilePath);
        if (!$xmlDoc->schemaValidate(DatastructsSchema)) { return false; };

        //load Root-Element
        $datastruct = simplexml_load_file($DatastructFilePath);

        if($datastruct !== false){

            //get Table und Columns Classes
            $tableFile = realpath(ScriptsDirectory . 'Table.php');
            $columnsFile = realpath(ScriptsDirectory . 'Column.php');
            if(!($tableFile === false || $columnsFile === false)){
                require_once $tableFile;
                require_once $columnsFile;

                $UOTableArray = array();

                //iterate through Tables
                foreach($datastruct as $XMLTableTag=>$XMLTable){
                    if($XMLTableTag != "Table"){ continue; };

                    $tableName = $XMLTable['TableName'];
                    $tableType = $XMLTable['Use'];
                    $tableDescGerm = $XMLTable->TableDescriptionGerman;
                    $tableDescEngl = $XMLTable->TableDescriptionEnglish;

                    $Table = new Table(
                        $tableName,
                        $tableType,
                        $tableDescGerm,
                        $tableDescEngl
                    );

                    foreach($XMLTable as $ColumnTag=>$AbstractColumn){
                        if($ColumnTag == "Column"){
                            $XMLColumn = $AbstractColumn->DataColumn[0];
                            $columnName = (string)$AbstractColumn['ColumnName'];
                            if(isset($XMLColumn)){
                                $ColumnDescGerm = (string)$XMLColumn->ColumnDescriptionGerman;
                                $ColumnDescEngl = (string)$XMLColumn->ColumnDescriptionEnglish;
                                $ColumnType = $XMLColumn->DataColumnType->children()[0];
                                $ColumnTypeName = (string)$ColumnType->getName();
                                $Column;

                                switch ($ColumnTypeName) {
                                    case "NVARCHAR": case "NCHAR":
                                        $length = (string)$ColumnType['Length'];
                                        $Column = new CharacterColumn(
                                            $columnName,
                                            $ColumnTypeName,
                                            $ColumnDescGerm,
                                            $ColumnDescEngl,
                                            $length
                                        );
                                        break;

                                    case "NUMERIC":
                                        $precision = (integer)$ColumnType['Precision'];
```

```
                                    $scale = (integer)$ColumnType['Scale'];
                                    $Column = new NumericColumn(
                                        $columnName,
                                        $ColumnTypeName,
                                        $ColumnDescGerm,
                                        $ColumnDescEngl,
                                        $precision,
                                        $scale
                                    );
                                    break;

                                default:
                                    $columnOtherTypeName = (string)$ColumnType['Type'];
                                    $Column = new Datacolumn(
                                        $columnName,
                                        $columnOtherTypeName,
                                        $ColumnDescGerm,
                                        $ColumnDescEngl
                                    );
                                    break;
                            }

                            foreach($XMLColumn as $Tag=>$XMLConstraint){
                                if($Tag == "Constraint"){
                                    $constraintType = $XMLConstraint->children()[0];
                                    $constraintTypeName = (string)$constraintType->getName();
                                    if($constraintTypeName == "SimpleConstraint"){
                                        $simpleConstraintType = (string)$constraintType['Type'];
                                        if($simpleConstraintType == "PRIMARY KEY"){
                                            $Column->setPrimaryKey();
                                        }
                                    }
                                    else{
                                        $refTable = (string)$constraintType['RefTableName'];
                                        $refColumn = (string)$constraintType['RefColumnName'];
                                        $Column->setForeignKey($refTable, $refColumn);
                                    };
                                }
                            }
                            $Table->addColumn($Column);
                        }
                       else{
                            $XMLSpatialColumn = $AbstractColumn->SpatialColumn;
                            if(isset($XMLSpatialColumn)){
                                $WKTColumn = new Datacolumn(
                                    '_WKT_' . $columnName,
                                    'TEXT',
                                    'Geometry in WKT-Format',
                                    'Geometry in WKT-Format'
                                );
                                $Table->addColumn($WKTColumn);
                            };
                        }//End if SpatialColumn
                    };//End if Column
                }//End foreach Column in Tables

                $UOTableArray[$Table->getName()] = $Table;

            }
            if(count($UOTableArray) > 0){ return $UOTableArray; }
            else{ return false; }
        }
        else{  return false; }
    }
    else{ return false; }
}

//****************************************************************************
//                      Functions: sort Array of Tables
//****************************************************************************

    //give Tables their Ranks
    function recArraySorter(array &$Array, $tableName){
            $table = $Array[$tableName];
            if($table->getType() === "Vocabulary"){ $table->setRank(0); }
            else if($table->referencesTables()){
                $Rank = 1;
                $onlyGlossRef = true;
                foreach ($table->getReferencedTables() as $refTableName) {
                    $refTable = $Array[$refTableName];
                    if($refTable->getType() === "Storage"){ $onlyGlossRef = false; };
                    if(!$refTable->rankIsSet()){ recArraySorter($Array, $refTableName); };
                    $refRank = $refTable->getRank();
                    if($refRank > $Rank){ $Rank = $refRank; };
                }
                if($onlyGlossRef){ $table->setRank(1); }
                else{ $table->setRank($Rank + 1); }
            }
            else{ $table->setRank(1); }
```

```
    };

    //Sort-Function for usort()
    function userSort(Table &$a, Table &$b){
        $aRank = $a->getRank();
        $bRank = $b->getRank();
        if($aRank < $bRank){ return -1; }
        else if($aRank > $bRank){ return 1; }
        else { return 0; };
    }

//*****************************************************************************
//                          Main-Function
//*****************************************************************************

    function createDatastructHTMLFile($datastructName){

        //Path to Datastructure
        $DatastructFilePath = DatastructsDirectory . $datastructName . '.xml';

        //generate Output-String
        $htmlText =
            "<div class=\"datastruct_name\" id=\"datastruct_name\">Databasestructure</div>\n"
            . "<ul class=\"datastruct_tableslist\" id=\"datastruct_tableslist\">\n";

        //generate Tables
        $tables = generateTablesFromXMLDatastructure($datastructName);

        //open File
        $ContentsFile = fopen(DatastructsHTMLDirectory . $datastructName . '.html', "w");

        if($tables !== false){
            $root = simplexml_load_file($DatastructFilePath);

            //generate Output-String
            $htmlText =
                "<div id=\"datastruct_name\">" . htmlentities($datastructName) . "</div>\n"
                . "<ul id=\"datastruct_tableslist\">\n";


            //iterate over Tables and set Ranks
            $keys = array_keys($tables);
            $keysLength = count($keys);
            for($index = 0; $index < $keysLength; $index++){
                recArraySorter($tables, $tables[$keys[$index]]->getName());
            };

            //sort Array according to Ranks
            usort($tables, "userSort");

            //write to File
            $currentRank = 0;
            if($ContentsFile !== false){

                $storageLabelSet = false;

                fwrite($ContentsFile, $htmlText);
                if($tables[0]->getRank() === 0){
                    fwrite($ContentsFile, "<li class=\"datastruct_level\">Glossaries</li>\n");
                }
                else{
                    fwrite($ContentsFile, "<li class=\"datastruct_level\">-----</li>\n");
                    $storageLabelSet = true;
                }

                foreach ($tables as $table) {
                    if($table->getRank() > $currentRank){
                        $currentRank = $table->getRank();
                        if($storageLabelSet === true){ fwrite($ContentsFile, "<li class=\"datastruct_level\">-----</li>\n"); }
                        else{ $storageLabelSet = true; fwrite($ContentsFile, "<li class=\"datastruct_level\">Tables</li>\n"); };
                    }
                    fwrite($ContentsFile, $table->ToHTML());
                }
                fwrite($ContentsFile, "\t</ul>");
                fclose($ContentsFile);
                return true;
            }
            else{
                return false;
            };
        }
        else{
            $htmlText .= '\n</ul>';
            fwrite($ContentsFile, $htmlText);
            fclose($ContentsFile);
            return true;
        }
    }
?>
```

MAINCONNECTIONUTILITY.PHP

```php
<?php

//*****************************************************************************
//  SQLite-Extensions: configure PHP_INI_SYSTEM
//  Windows: php_sqlite3 must be enabled manually
/*
     Activate SQLITE3:
          set in: php.ini:
               ...
               extension=php_sqlite3.dll
               ...
               [sqlite3]
               sqlite3.extension_dir = "./ext"
*/
//*****************************************************************************

//*****************************************************************************
//                          Return-Values-Array
//*****************************************************************************

$RETURNVALUES = array(
    0=>true,
    1=>'Invalid Parameter',
    2=>'Cannot find Master-Database',
    3=>'No Result',
    4=>'Repository not found',
    5=>'Invalid Repository-Name',
    6=>'Invalid Datastructure-name',
    7=>'No Tablenames specified',
    8=>'Invalid Tablename(s)',
    9=>'Selected Columns missing',
    10=>'Invalid Select-String'
);

//*****************************************************************************
//                             Utilities
//*****************************************************************************

    //get current Directory
    define('CurrentDirectory', __DIR__);
    define('RessourceDirectory', dirname(CurrentDirectory));
    define('MasterDBPath', RessourceDirectory . DIRECTORY_SEPARATOR . "MasterDB" . DIRECTORY_SEPARATOR . 'MasterDB.sqlite');

//*****************************************************************************
//                        Establish SQLite-Connection
//*****************************************************************************

    //Connection-Object
    class SQLiteDBConn extends SQLite3{
        function __construct($path){ $this->open($path, SQLITE3_OPEN_READONLY); }
        function __destruct(){ $this->close(); }
    }

//*****************************************************************************
//                                  get Repository-Path
//*****************************************************************************

    function getRepositoryPath($DatastructName, $RepositoryName){
          if(isset($DatastructName) && is_string($DatastructName) && $DatastructName !== '' &&
             isset($RepositoryName) && is_string($RepositoryName) && $RepositoryName !== ''){
               if(file_exists(MasterDBPath)){
                    $dbHandle = new SQLiteDBConn(MasterDBPath);
                    if(isset($dbHandle)){
                         str_replace("'", "", $RepositoryName);
                         str_replace("'", "", $DatastructName);
                         $result = $dbHandle->querySingle("SELECT AbsolutePath from Repositories WHERE Datastruct='" .
                           $DatastructName .
                                 "' AND RepositoryName='" . $RepositoryName . "'", false);
                         if($result === false || $result === null){ return 3; }
                         else if(realpath($result) === false){ return 4; }
                         else { return $result; }
                    }
                    else{ return 2; };
               }
               else{ return 2; };
          }
          else{ return 1; }
    }


//*****************************************************************************
//                          get Repositories as HTML
//*****************************************************************************

    //function: create RepositoriesHTML, if not exists
     function getRepositoriesHTML($datastructName){
```

```
        $dbHandle = new SQLiteDBConn(MasterDBPath);
        if(isset($dbHandle)){
            $html = "";
            $repNamesResult = $dbHandle->query("SELECT RepositoryName from Repositories WHERE Datastruct='" . $datastructName .
                "'");
            $checkedValue = " checked";
            while(true){
                $row = $repNamesResult->fetchArray();
                if($row === false){ break; }
                else{
                    $repname = $row['RepositoryName'];
                    $html .= "<li><input type='radio' name='Repository' value='$repname'$checkedValue/>" . htmlentities($repname,
                        ENT_QUOTES) . "</li>";
                    $checkedValue = "";
                };
            };
            $repNamesResult->finalize();
            return $html;
        }
        else{ return '<p>' . $RETURNVALUES[2] . '</p>'; };
    };

//*****************************************************************************
//                          get Repository Names
//*****************************************************************************

    //function: create RepositoriesHTML, if not exists
     function getRepositoryNames($datastructName){
        $dbHandle = new SQLiteDBConn(MasterDBPath);
        $repositories = array();
        if(isset($dbHandle)){
            $repNamesResult = $dbHandle->query("SELECT RepositoryName from Repositories WHERE Datastruct='" . $datastructName .
                "'");
            $checkedValue = " checked";
            while(true){
                $row = $repNamesResult->fetchArray();
                if($row === false){ break; }
                else{ $repositories[] = $row['RepositoryName']; };
            };
            $repNamesResult->finalize();
        };
        return $repositories;
    };


?>
```

MAINJSSCRIPT.JS

```
//*****************************************************************************
//                          make global Variable
//*****************************************************************************

var cy = null;

//*****************************************************************************
//                    register Event-Handlers and Functions
//*****************************************************************************

$(function() {

    //*************************************************************************
    //                set Glossary-Tables in a different Color
    //*************************************************************************

        $(".table_is_glossary").parent(".datastruct_table").css('background-color', '#747272');
        $(".table_is_glossary").siblings('.datastruct_columns_list').css('background-color', '#5E5D5D');

    //*************************************************************************
    //                    register Event-Handlers and Functions
    //*************************************************************************

        //*********************************************************************
        //                  Event-Handler for Setting Query's From-Tables
        //*********************************************************************
        (function(){
                function orderFunction(columnIDNew, qualifiedName, columnCategory){
                        if(columnCategory === 'TEXT' || columnCategory === 'NUMBER' || columnCategory === 'BOOLEAN'){
                                var columnItem = $("#" + columnIDNew);
                                  columnItem.children(".selected_Column_checkbox").change(function(){
                                          var checkBox = $(this);
                                          if(checkBox.is(":checked")){
                                                  var item = "<li id='order_" + columnIDNew.substring(7) + "'><span>" + qualifiedName +
                                                      "</span><select><option>ASC</option><option>DESC</option></select><button
                                                        type='button'>X</button><br></li>";
                                                  $("#orderBy_list").append(item);
                                                  $("#order_" + columnIDNew.substring(7) +
                                                        ">button").click(function(){ $(this).parent("li").remove(); });

                                                  //add Aggregate
                                                  if(columnCategory === 'TEXT' || columnCategory === 'DATE'){
                                                          columnItem.children("span").after("<select
                                                        class='selCol_aggregate'><option>none</option><option>count</option>
                                                       </select>"); }
                                                  else if(columnCategory === 'NUMBER'){
                                                          columnItem.children("span").after("<select
                                                        class='selCol_aggregate'><option>none</option><option>count</option>
                                                       <option>min</option><option>max</option><option>avg</option>
                                                       <option>sum</option><option>total</option></select>");
                                                  }

                                                  var aggregateSelect =  $("#" + columnIDNew + ">select.selCol_aggregate");

                                                  //set aggregate hidden, if another aggregate is set
                                                  if($("select.selCol_aggregate>option:selected:not(:contains('none'))").length){
                                                        aggregateSelect.hide(); }

                                                  //add event handler for Aggregate function
                                                  aggregateSelect.change(function(){
                                                          if($(this).val() === 'none'){
                                                                  $("#orderBy_container>h2").text('Order by');
                                                                  checkBox.show();
                                                                  $("select.selCol_aggregate").show();
                                                          }
                                                          else{
                                                                  var orderColumn = $("#order_" + columnIDNew.substring(7));
                                                                  if(orderColumn.length){ orderColumn.remove(); }
                                                                  $("#orderBy_container>h2").text('Group by / Order by');
                                                                  checkBox.hide();
                                                                  $("select.selCol_aggregate>option:selected:contains('none')")
                                                                          .each(function(){
                                                                          $(this).parent("select.selCol_aggregate").hide();
                                                                  });
                                                          }
                                                  });
                                          }
                                          else{
                                                  var sortColumn = $("#order_" + columnIDNew.substring(7));
                                                  if(sortColumn.length){ sortColumn.remove(); }
                                                  var aggregate = $("#" + columnIDNew + ">select.selCol_aggregate");
                                                  if(aggregate.length){
                                                          aggregate.val('none');//triggers change event
                                                          aggregate.remove();
                                                  }
                                          }
                                  });
                        }
                }

                function setQueryFromTable(fromTableName, columnsLink){

                    if(!$("#from_tables_list>li[id = 'from_" + fromTableName + "']").length){

                        var tableLink = arguments.length > 1 ? ("<p>" + arguments[1] + "</p>") : "";
                        var fromTableItem = "<li class='from_table' id='from_" + fromTableName + "'><p>" + fromTableName + "</p>" +
                           tableLink;
```

```
//get all Columns
$("#table_" + fromTableName + " .datastruct_column").each(function(){
        var currentColumn = $(this);
        var columnID = currentColumn.attr("id");
        var columnName = columnID.substring(7, columnID.length-fromTableName.length-1);
        var isGeom = false;
        if(columnName.startsWith('_GML_') || columnName.startsWith('_WKT_')){ isGeom = true; }
        var columnNameDisplay = fromTableName + "." + columnName;
        var columnIDNew = "selCol_" + columnID.substring(7);
        var columnCategory = currentColumn.children(".datastruct_column_cat").text();

        //check ColumnType
        if(columnCategory !== 'BLOB' && columnCategory !== 'GEOMETRY'){
                var item  = "<li class='selected_Column' id='" + columnIDNew +
                        "'><input type='checkbox' class='selected_Column_checkbox' value='query'/><span
                     class='selCol_name'>" + columnNameDisplay + "</span></li>";

                $("#selected_columns_list").append(item);

                $("#" + columnIDNew + ">select.selCol_aggregate").hide();

                if(!isGeom && columnCategory === 'TEXT'){
                        $("#" + columnIDNew + ">span").click(function(){
                                var currentSelColumn = $(this);
                                var conditionBlock = currentSelColumn.siblings("p");
                                if(conditionBlock.length){ conditionBlock.remove(); }
                                else{
                                        var newConditionBlock = "<p class='cond_text'><select
                                        class='cond_operator'><option>=</option><option>LIKE</option>
                                        </select><input class='cond_literal' type='text'></p>";
                                        currentSelColumn.parent(".selected_Column").append(newConditionBlock);
                                }
                        });
                        orderFunction(columnIDNew, columnNameDisplay, columnCategory);
                }
                else if(columnCategory === 'NUMBER'){
                        $("#" + columnIDNew + ">span").click(function(){
                                var currentSelColumn = $(this);
                                var conditionBlock = currentSelColumn.siblings("p");
                                if(conditionBlock.length){ conditionBlock.remove(); }
                                else{
                                        var newConditionBlock = "<p class='cond_numeric'><select
                                        class='cond operator'><option>=</option><option><</option><option>>
                                        </option></select><input class='cond_literal' type='number'
                                        step='0.0001'></p>";
                                        currentSelColumn.parent(".selected_Column").append(newConditionBlock);
                                }
                        });
                        orderFunction(columnIDNew, columnNameDisplay, columnCategory);
                }
                else if(columnCategory === 'BOOLEAN'){
                        $("#" + columnIDNew + ">span").click(function(){
                                var currentSelColumn = $(this);
                                var conditionBlock = currentSelColumn.siblings("p");
                                if(conditionBlock.length){ conditionBlock.remove(); }
                                else{
                                        var newConditionBlock = "<p class='cond_bool'><span>=</span><select
                                        class='cond_bool_literal'><option>true</option><option>false
                                        </option></select></p>";
                                        currentSelColumn.parent(".selected_Column").append(newConditionBlock);
                                }
                        });
                        orderFunction(columnIDNew, columnNameDisplay, columnCategory);
                }
        }
});

var listItems = "";

//get all Tables referenced by this Table
$("#table " + fromTableName + " .datastruct_tableFK").each(function(){
        var thisColumn = $(this);
        var sourceColName = thisColumn.siblings(".datastruct_column_name").text();
        var targetTable = thisColumn.children("span").attr("href").substring(7);

        if($("#from_tables_list>li[id='from_" + targetTable + "']").length === 0){
            var targetColumnID = thisColumn.siblings(".datastruct_columnFK").children("span").attr("href");
            var targetColumn = targetColumnID.substring(8, targetColumnID.length-targetTable.length-1);

            listItems += "<li class='from_ref_" + targetTable + "'>";
            listItems += "<p>" + targetTable + "</p>";
            listItems += "<p>" + fromTableName + "." + sourceColName + " = " +  targetTable + "." +
            targetColumn + "</p>";
            listItems += "</li>";
        }
});

//get all Tables referencing this Table
$(".datastruct_tableFK>span[href='#table_" + fromTableName + "']").each(function(){
        var spanElement = $(this);
        var targetColumnID =
            spanElement.parent().siblings(".datastruct_columnFK").children("span").attr("href");
        var targetColumnName = targetColumnID.substring(8, targetColumnID.length-fromTableName.length-1);
        var sourceTableName = spanElement.parents(".datastruct_table").attr("id").substring(6);
        var sourceColumnID = spanElement.parents(".datastruct_column").attr("id");
        var sourceColumnName = sourceColumnID.substring(7, sourceColumnID.length-sourceTableName.length-1);

        if($("#from_tables_list>li[id = 'from_" + sourceTableName + "']").length === 0){
            listItems += "<li class='from_ref_" + sourceTableName + "'>";
            listItems += "<p>" + sourceTableName + "</p>";
            listItems += "<p>" + sourceTableName + "." + sourceColumnName + " = " +  fromTableName + "." +
            targetColumnName + "</p>";
            listItems += "</li>";
```

```
                        }
                });

                if(listItems.length){ fromTableItem += "<ul>" + listItems + "</ul>"; }

                fromTableItem += "</li>";

                $("#from_tables_list").append(fromTableItem);

                //add event-Handler for related Tables
                $(".from_table > ul > li").click(function(){
                                var currentItem = $(this);
                                var tableName = currentItem.children("p:first-child").text();
                                var connection = currentItem.children("p:last-child").text();
                                setQueryFromTable(tableName, connection);
                                $(".from_ref_" + tableName).remove();
                        }
                );

                return true;
            }
            else{ return false; }
        }

        $("#from_tables_container > button[value = 'clear']").hide();
        $("#orderBy_container").hide();
        $("#selected_columns_container>h2").hide();
        $("#limit_offset_send_container").hide();

        //set first from-Table
        $("#from_tables_container > button[value = 'setFirstFromTable']").click(function(){
                        var currentTableOption =
                             $("#datastruct_panel01>select[name='datastruct_panel_tables']>option").filter(":selected");
                        var currentTableName = currentTableOption.val().substring(6);
                        var firstTableSet = setQueryFromTable(currentTableName);
                        if(firstTableSet){
                                $(this).hide();
                                $("#from_tables_container > button[value = 'clear']").show();
                                $("#orderBy_container").show();
                                $("#selected_columns_container>h2").show();
                                $("#limit_offset_send_container").show();
                        }
                }
        );

});

//clear from-Tables
$("#from_tables_container > button[value = 'clear']").click(function(){
                $(".from_table").remove();
                $("#selected_columns_list>li").remove();
                $("#orderBy_list>li").remove();
                $("#from_tables_container > button[value = 'setFirstFromTable']").show();
                $(this).hide();
                $("#orderBy_container").hide();
                $("#selected_columns_container>h2").hide();
                $("#limit_offset_send_container").hide();
                $("#orderBy_container>h2").text("Order by");
        }
);

//*****************************************************************************
//                Event-Handler for preparing and sending Queries
//*****************************************************************************
$("#query_results_area").hide();
$("nav#main_menu>ul>li:nth-child(3)").hide();
$("#query_results_area>button:nth-child(3)").hide();
$("#query_results_area>button:nth-child(4)").hide();

function setQueryResults(data){
        $("#query results container").html(data);
        $("#query_results_area>button:nth-child(3)").hide();
        $("#query_results_area>button:nth-child(4)").hide();
        var unselectedRows = $("#query_results_container table tr.unselected");
        unselectedRows.click(function(){
                var layerSelected = $("#map_layers_manager>select[name='current_vectorlayer']>option").length > 0;
                var tableRow = $(this);
                if(tableRow.attr('class') === 'unselected'){tableRow.attr('class', 'selected'); }
                else if(tableRow.attr('class') === 'selected'){ tableRow.attr('class', 'unselected'); }

                var selectedCount = $("#query_results_container table tr.selected").length;
                if(selectedCount > 0 && layerSelected){ $("#query_results_area>button:nth-child(3)").show(); }
                else{ $("#query_results_area>button:nth-child(3)").hide(); }
                if(selectedCount === 1){ $("#query_results_area>button:nth-child(4)").show(); }
                else{ $("#query_results_area>button:nth-child(4)").hide(); }
        });
        unselectedRows.mouseover(function(event){
                if(event.buttons===1){
                        var layerSelected = $("#map_layers_manager>select[name='current_vectorlayer']>option").length > 0;
                        var tableRow = $(this);
                        if(tableRow.attr('class') === 'unselected'){ tableRow.attr('class', 'selected'); }
                        else if(tableRow.attr('class') === 'selected'){ tableRow.attr('class', 'unselected'); }

                        var selectedCount = $("#query_results_container table tr.selected").length;
                        if(selectedCount > 0 && layerSelected){ $("#query_results_area>button:nth-child(3)").show(); }
                        else{ $("#query_results_area>button:nth-child(3)").hide(); }
                        if(selectedCount === 1){ $("#query_results_area>button:nth-child(4)").show(); }
                        else{ $("#query_results_area>button:nth-child(4)").hide(); }
                }
        });
}
function errorFunction(){ $("#query_results_container").html("<p>Sorry, an Error occurred!</p>"); }
```

```
$("#limit offset send container>button").click(function(){
        var datastruct = $("#datastruct_name").text();
        var repository = $("#repositories_list input[name='Repository']:checked");
        if(!repository.length){ return; }
        var repositoryName = repository.val();
        var fromTablesCount = $(".from_table").length;
        var selectedColumns = $(".selected_Column").has("input[class='selected_Column_checkbox']:checked");

        if(fromTablesCount>0 && selectedColumns.length>0){

                var limitOffestContainer = $("#limit_offset_send_container");
                limitOffestContainer.hide();
                var clearButton = $("#from_tables_container>button[value='clear']");
                clearButton.hide();

                //get 'FROM' Line
                var fromString = $(".from_table:nth-of-type(1)").attr("id").substring(5) + ";";
                for(var i = 2; i <= fromTablesCount; i++){
                        var tableToJoin = $(".from_table:nth-of-type(" + i + ")");
                        var joinedColumns = tableToJoin.children("p:nth-child(2)").text();
                        var joinStmt = tableToJoin.attr("id").substring(5) + "(" + joinedColumns + ");";
                        fromString = fromString + joinStmt;
                }
                fromString = fromString.substring(0, fromString.length-1);

                //get all selected Columns and Aggregates
                var selectedColumnsString = "";
                var aggregateString = "";
                var aggregate = $("select.selCol_aggregate>option:selected:not(:contains('none'))");

                if(aggregate.length){
                        aggregateString = aggregate.val();
                        selectedColumnsString += aggregate.parent("select").siblings("span.selCol_name").text();
                }
                else{
                        selectedColumns.each(function(){
                                selectedColumnsString += $(this).children("span.selCol_name").text() + ";";
                        });
                        selectedColumnsString = selectedColumnsString.substring(0, selectedColumnsString.length-1);
                }

                //get all Columns' conditions
                var conditionsString = "";
                $(".selected_Column:has(select.cond_operator)").each(function(){
                        var currentColumn = $(this);
                        var columnName = currentColumn.children("span.selCol_name").text();
                        var operator = currentColumn.children("p").children("select.cond_operator").val();
                        var literal = currentColumn.children("p").children("input.cond_literal").val();
                        literal = literal.replace(/'/g, "\\'");
                        if(operator === 'LIKE' && !literal.includes('%')){ literal = '%' + literal + '%'; }
                        if(currentColumn.children("p.cond_text").length){ literal = "'" + literal + "'"; }
                        conditionsString += columnName + ';' + operator + ';' + literal + ";";
                });
                $(".selected_Column:has(select.cond_bool_literal)").each(function(){
                        var currentColumn = $(this);
                        var columnName = currentColumn.children("span.selCol_name").text();
                        var literal = currentColumn.children("p").children("select.cond_bool_literal").val();
                        conditionsString += columnName + ';=;' + literal + ";";
                });
                if(conditionsString.length > 0){
                     conditionsString = conditionsString.substring(0, conditionsString.length-1);
                }

                //get ordering Columns
                var orderString = "";
                var groupByString = "";
                $("#orderBy_list>li").each(function(){
                        var currentColumn = $(this);
                        var columnName = currentColumn.children("span").text();
                        if(aggregateString.length){ groupByString += columnName + ";"; }
                        var order = currentColumn.children("select").val();
                        orderString += columnName + ";" + order + ";";
                });
                if(groupByString.length > 0){
                     groupByString = groupByString.substring(0, groupByString.length-1);
                }
                if(orderString.length > 0){
                     orderString = orderString.substring(0, orderString.length-1);
                }

                var limit = $("#limit_offset_send_container>input[name='query_limit']").val();

                var offset = $("#limit_offset_send_container>input[name='query_offset']").val();

                $("#query results container>table").remove();
                $("#query_results_container>p").remove();
                $("#query_results_container").html("<p>retrieving data... please wait</p>");

                var data = {
                        datastruct: datastruct,
                        repository: repositoryName,
                        aggregate: aggregateString,
                        fromString: fromString,
                        selectedColumnsString: selectedColumnsString,
                        orderString: orderString,
                        conditionsString: conditionsString,
                        groupByString: groupByString,
                        sqlLimit: limit,
                        sqlOffset: offset
                };

                $.ajax({
                        url: "http://" + $(window.location).attr('hostname') + "/Query.php",
```

```
                                type: "POST",
                                data: data,
                                dataType : 'html',
                                timeout: 20000,
                                error: errorFunction,
                                success: setQueryResults,
                                complete: function(){ limitOffestContainer.show(); clearButton.show(); }
                        }
                );

                $("#query_results_area").show();
                $("nav#main_menu>ul>li:nth-child(3)").show();

                $("html, body").animate({
                        scrollTop : $("#query_results_area").offset().top },
                        1000
                );
        }
});



//*****************************************************************************
//                  Event-Handler for Datastruct-Tables-Dropdown
//*****************************************************************************

    $(".datastruct_table>p:first-child").click(function(event){
        if(event.target === this){
            var table = $(event.target);
            var tableID = table.parent().attr("id");
            var colListID = "#" + tableID + "_columns";
            var tablesDropDown = $('select[name = "datastruct_panel_tables"]');
            if(table.siblings("ul").is(":visible")){
                var topValue = $('select[name = "datastruct_panel_tables"]>option:first-child').val();
                tablesDropDown.val(topValue);
                selectHelper(null); }
            else{
                var node = cy.$('#' + tableID.substring(6));
                selectHelper(node);
                tablesDropDown.val(tableID);
            }
            $(colListID).slideToggle("slow");
        }
    });

//*****************************************************************************
//                  Event-Handler for Jumping from FK to Table/Column
//*****************************************************************************

    $(".datastruct_FK>span[href]").click(function(){
        var target = $(this);
        var href = target.attr("href");
        var parent = $('#datastruct');

        var wantedTableName;
        var expand = false;

        if(href.startsWith('#table_')){ wantedTableName = href; }
        else{ expand = true; wantedTableName = target.parent().siblings(".datastruct_tableFK").children("span").attr("href"); }

        var wantedTable = $(wantedTableName);
        var wantedTableNameItem = wantedTable.children("p:first-child");
        var wantedElement = $(href);

        var columnsList = wantedTable.children("ul");

        if(expand){ if(expand && columnsList.is(":visible") === false){ wantedTableNameItem.trigger('click'); } }
        else{
                var topValue = $('select[name = "datastruct_panel_tables"]>option:first-child').val();
                $('select[name = "datastruct_panel_tables"]').val(topValue);
                selectHelper(null);
        }

        parent.animate({
            scrollTop : parent.scrollTop() + wantedElement.position().top },
            'slow'
        );

        target.parents(".datastruct_columns_list").slideToggle("slow");
    });


//*****************************************************************************
//      Event-Handler for Assigning LineWidth and Text-Height to Diagram
//*****************************************************************************

    $('select[name="datastruct_panel_linewidth"]').on('change', function(){
        cy.style().selector('edge').style( {'width' : $(this).val() }).update();
    });

    $('select[name="datastruct_panel_fontheight"]').on('change', function(){
        cy.style().selector('node').style( {'font-size' : $(this).val() }).update();
    });

//*****************************************************************************
//              Event-Handler selecting Table by Dropdown-Menu
//*****************************************************************************

    $('select[name = "datastruct_panel_tables"]').change(function(event){
         if(event.target === this){
            var datastruct = $('#datastruct');
            var table = $(this).val().substring(6);
            var wantedTable = $('#table_' + table);
```

```
                datastruct.animate( { scrollTop : datastruct.scrollTop() + wantedTable.position().top }, 'slow' );
                if(parseInt(wantedTable.siblings('.table_rank').text()) === 0){ selectHelper(null); }
                else{ selectHelper(cy.$('#' + table)); }
            }
        });

    //*************************************************************************
    //              Event-Handler selecting Table by Dropdown-Menu
    //*************************************************************************

        $('select[name = "datastruct_panel_columns"]').change(function(event){
            if(event.target === this){
                var datastruct = $('#datastruct');

                var selected = $('select[name = "datastruct_panel_columns"] :selected');
                var columnID = selected.val();
                var tableName = selected.attr('title');

                var wantedTable = $('#table_' + tableName);
                var columnsList = wantedTable.children('ul');
                var column = $('#' + columnID);

                if(columnsList.is(":visible") === false){
                    wantedTable.children('p:first-child').trigger('click');
                }
                datastruct.animate( { scrollTop : datastruct.scrollTop() + column.position().top }, 'slow' );
            }
        });


//*************************************************************************
//     declare Variables for Cytoscape-Diagram and Datastruct-Diagram
//*************************************************************************

    (function(){
            var cyElements = [];
            var dropDownTableNames = [];
            var dropDownColumnNames = [];

        //*************************************************************************
        //                  set Datastruct-Diagram-px-Dropdowns
        //*************************************************************************

            function setDatastructPX(){
                var pxLinewidth = $('select[name="datastruct_panel_linewidth"]');
                var pxFontHeight = $('select[name="datastruct_panel_fontheight"]');
                for(var i = 1; i < 101 ; i++){
                    pxLinewidth.append($("<option></option>").attr("value", i).text(i));
                    pxFontHeight.append($("<option></option>").attr("value", i).text(i));
                }
                $('select[name="datastruct_panel_linewidth"]>option[value="5"]').attr('selected', 'selected');
                $('select[name="datastruct_panel_fontheight"]>option[value="10"]').attr('selected', 'selected');
            }
            setDatastructPX();

        //*************************************************************************
        //    Loop through Tables and Columns for Cytoscape and Datastruct-Dropdowns
        //*************************************************************************

            //get Diagram-Nodes aka Tables
            $(".datastruct_table").each(function(){
                var tableItem = $(this);
                var tableIDFull = tableItem.attr('id');
                dropDownTableNames.push(tableIDFull);
                var tableID = tableIDFull.substring(6);
                var tableRank = parseInt(tableItem.children('.table_rank').text());
                if(tableRank > 0){
                    var tableType = "NOTROOT";
                    if(tableRank === 1){ tableType = "ROOT"; }
                    var node = { data: { id: tableID, type: tableType } };
                    cyElements.push(node);
                }

            });

            //get Cytoscape edges
            $(".datastruct_tableFK").each(function(){
                var FKItem = $(this);
                var tableID = FKItem.parents(".datastruct_table").attr('id').substring(6);
                var href = FKItem.children("span").attr('href');
                var fkTable = href.substring(7, href.length);
                var edge = { data: { id: tableID + ':TO:' + fkTable, source: tableID, target: fkTable, colored: 'false' } };
                cyElements.push(edge);
            });

            //set Datastruct-Tables-Dropdown
            dropDownTableNames.sort();
            var tablesCount = dropDownTableNames.length;
            var tablesSelect = $('select[name="datastruct_panel_tables"]');
            var columnSelect = $('select[name="datastruct_panel_columns"]');
            for(var tableNr = 0; tableNr < tablesCount; tableNr++){
                tablesSelect.append($("<option></option>").attr("value",
                        dropDownTableNames[tableNr]).text(dropDownTableNames[tableNr].substring(6)));
            }

            //get Column-Names
            $('.datastruct_column').each(function(){ dropDownColumnNames.push($(this).attr('id')); });

            //set Datastruct-Columns
            dropDownColumnNames.sort();
            var columnCount = dropDownColumnNames.length;
            for(var colNr = 0; colNr < columnCount; colNr++){
                var columnID = dropDownColumnNames[colNr];
```

```
            var column = $('#' + columnID);
            var columnText = column.children('.datastruct_column_name').text();
            var tableName = columnID.substring(7 + columnText.length + 1);
            columnSelect.append($("<option></option>").attr("value", columnID).attr("title", tableName).text(columnText));
        }


//*************************************************************************
//                                Cytoscape Diagram
//*************************************************************************

    //*************************************************************************
    //                      Make different Cytoscape-Layouts
    //*************************************************************************

    var circleLayout = {
        name: 'circle',
        sort: function(a, b){
            var tableARank = parseInt($("#table_" + a.id()).siblings(".table_rank").text());
            var tableBRank = parseInt($("#table_" + b.id()).siblings(".table_rank").text());
            var returnValue = tableBRank-tableARank;
            if(returnValue === 0){
                var tableIdA = 'span[href^="#table_' + a.id() + '"]';
                var FKCountA = $(".datastruct_columnFK").children(tableIdA).length;
                var tableIdB = 'span[href^="#table_' + b.id() + '"]';
                var FKCountB = $(".datastruct_columnFK").children(tableIdB).length;
                if(FKCountA > FKCountB){ return 1; }
                else if(FKCountA < FKCountB){ return (-1); }
                else { return 0; }
            }else{ return tableARank-tableBRank; }
        },
        animate: true,
        animationDuration: 3000,
        fit: true
    };

    var concentricLayout = {
        name: 'concentric',
        concentric: function(node){
            var tableId = 'span[href^="#table_' + node.id() + '"]';
            var FKCount = $(".datastruct_tableFK").children(tableId).length;
            return FKCount;
        },
        levelWidth: function(nodes){ return 1; },
        animate: true,
        animationDuration: 3000,
        fit: true
    };

    var breadthfirst = {
        name: 'breadthfirst',
        animate: true,
        animationDuration: 3000,
        fit: true
    };

    var breadthfirst_roots = {
        name: 'breadthfirst',
        roots: 'node[type = "ROOT"]',
        animate: true,
        animationDuration: 3000,
        fit: true
    };

    //*************************************************************************
    //                Event-Handler for Assigning Layout to Diagram
    //*************************************************************************

    $('#datastruct_panel02>button').on('click', function(){
        var lineWidth = $('select[name="datastruct_panel_linewidth"]');
        var fontHeight = $('select[name="datastruct_panel_fontheight"]');
        lineWidth.val(5);
        fontHeight.val(10);
        lineWidth.trigger('change');
        fontHeight.trigger('change');

        var value = $(this).val();

        if(value === "concentric"){ cy.layout(concentricLayout); }
        else if(value === "breadthfirst"){ cy.layout(breadthfirst); }
        else if(value === "breadthfirst_roots"){ cy.layout(breadthfirst_roots); }
        else{ cy.layout(circleLayout); }
    });

    //*************************************************************************
    //                          Make Cytoscape-Instance
    //*************************************************************************

    cy = cytoscape({

        container: document.getElementById('cy'),

        elements: cyElements,

        style: [ // the stylesheet for the graph
            {
                selector: 'node',
                style: {
                    'shape': 'rectangle',
                    'width': 'label',
                    'height': 'label',
                    //'padding': '40px',
                    'border-width': '2px',
```

```
                    'border-color': '#4d0000',
                    'background-color': '#4d0000',
                    'color': '#ff9933',
                    'text-valign': 'center',
                    'text-hlign': 'center',
                    'font-size': '10px',
                    'label': 'data(id)'
                }
            },

            {
                selector: 'node:selected',
                style: {
                    'color': '#ffff00'
                }
            },

            {
                selector: 'edge',
                style: {
                    'width': '5px',
                    'line-color': '#ffffff',
                    'curve-style': 'haystack',
                    'mid-target-arrow-color': '#ffffff',
                    'mid-target-arrow-shape': 'triangle-backcurve',
                    'mid-target-arrow-fill': 'filled'
                }
            },

            {
                selector: 'edge:selected',
                style: {
                  'line-color': '#CC8F33',
                }
            }
        ],

        layout: circleLayout,

        selectionType: 'single',

        motionBlur: true,

        wheelSensitivity: 0.2,

        boxSelectionEnabled: false

    });//End of Cytoscape-Definition


    //*************************************************************************
    //      Event-Handler for selecting Tables in the Diagram
    //*************************************************************************

    cy.on('tap', function(event){
        var node = event.cyTarget;

        var datastruct = $('#datastruct');
        var tablesDropDown = $('select[name = "datastruct_panel_tables"]');

        selectHelper(node);

        if(node !== cy && node.isNode()){
            //set Table in Dropdown
            tablesDropDown.val('table_' + node.id());

            //Scroll to Table in left Pane
            var wantedTable = $('#table_' + node.id());
            datastruct.animate( { scrollTop : datastruct.scrollTop() + wantedTable.position().top }, 'slow' );
        }
        else {
                var topValue = $('select[name = "datastruct_panel_tables"]>option:first-child').val();
                tablesDropDown.val(topValue);
                datastruct.animate( { scrollTop : 0 } );
        }
    });

})();

//*************************************************************************
//                          Helperfunctions
//*************************************************************************

    function selectHelper(node){
        //clear former Selections
        var currColored =  cy.elements('edge[colored = "true"]');
        currColored.animate({ style: { 'line-color': '#ffffff' } });
        currColored.data('colored', 'false');
        cy.nodes(':selected').unselect();
        //set new Selection
        if(node !== null && node !== cy && node.isNode()){
            node.select();
            var toEdges = node.connectedEdges('edge[target = "' + node.id() + '"]');
            var fromEdges = node.connectedEdges('edge[source = "' + node.id() + '"]');
            toEdges.animate({ style: { 'line-color': '#ff0000' } });
            fromEdges.animate({ style: { 'line-color': '#0066ff' } });
            toEdges.data('colored', 'true');
            fromEdges.data('colored', 'true');
        }
    }

//*************************************************************************
//                          Map-Functions
```

```
//*****************************************************************************

(function(){

	var vectorLayerCount = 0;
	var basemapCount = 1;
	var vectorLayers = [];
	var baseMapLayers = [];
	var vectorLayerStyles = [];

	var osmLayer = new ol.layer.Tile({
		source: new ol.source.OSM()
	});
	baseMapLayers.push(osmLayer);

	var view = new ol.View({
		projection: 'EPSG:3857',
		center: [0.0, 0.0],
		zoom: 1
	});

	var map = new ol.Map({
		target: 'map'
	});
	map.addLayer(osmLayer);
	map.setView(view);

	//*****************************************************************************
	//                              Style-Function
	//*****************************************************************************

	function styleFunctionAutomat(vectorLayerIndex, labels, fillColor, strokeColor, createNew){
		var style;
		var styleFunction;
		if(createNew){
			style = new ol.style.Style({
					stroke: new ol.style.Stroke({
						color: strokeColor,
						width: 1
					}),
					fill: new ol.style.Fill({ color: fillColor }),
					image: new ol.style.Circle({
						fill: new ol.style.Fill({ color: fillColor }),
						stroke: new ol.style.Stroke({
							color: strokeColor,
							width: 1
						}),
						radius: 4
					})
			});
			vectorLayerStyles[vectorLayerIndex] = style;
			styleFunction = function(feature){ return style; };

		}
		else if(fillColor && strokeColor){
			style = vectorLayerStyles[vectorLayerIndex];
			style.setFill(new ol.style.Fill({ color: fillColor }));
			style.setStroke(new ol.style.Stroke({
						color: strokeColor,
						width: 1
			}));
			style.setImage(new ol.style.Circle({
						fill: new ol.style.Fill({ color: fillColor }),
						stroke: new ol.style.Stroke({
							color: strokeColor,
							width: 1
						}),
						radius: 4
			}));
			styleFunction = function(feature){ return style; };
		}
		else{
			style = vectorLayerStyles[vectorLayerIndex];
			var oldstrokeColor = style.getStroke().getColor();
			if(labels){
				styleFunction = function(feature){
					style.setText(new ol.style.Text({
								font:'12px Calibri, sans-serif',
								fill: new ol.style.Fill({color: oldstrokeColor}),
								stroke: new ol.style.Stroke({
									color: oldstrokeColor,
									width: 0.5
								}),
								offsetY: 20,
								textBaseline: 'top',
								text: feature.get('name')
					}));
					return style;
				};
			}
			else{
				styleFunction = function(feature){
					style.setText(new ol.style.Text({
								font:'12px Calibri, sans-serif',
								fill: new ol.style.Fill({color: oldstrokeColor}),
								stroke: new ol.style.Stroke({
									color: oldstrokeColor,
									width: 0.5
								}),
								offsetY: 20,
								textBaseline: 'top',
								text: ''
					}));
```

```
                                return style;
                        };
                }
        }
        return styleFunction;
}

//************************************************************************
//                event-Handler for Creating a new Vectorlayer
//************************************************************************

$("#map_layers_manager>button[name='map_new_vectorlayer_button']").click(function(){
        var layerName = $("#map_layers_manager>input[name='map_new_vectorlayer_name']").val();
        if(layerName.length > 0){
                var layerIndex = vectorLayerCount;
                var layerNameComplete = (layerIndex + 1) + "-" + layerName;
                $("#map_layers_vector_list").append("<li id='" + layerIndex +
                        "vec'><input type='checkbox' id='map_layers_visibility_" + layerIndex + "' value='" +
                        layerIndex + "' checked/><select><option value='" + layerIndex + "' selected>---
                              </option><option value='" + layerIndex + "'>labels</option></select>" +
                              "<label>" + layerNameComplete + "</label></li>");

                var fillColor = '#' +
                    $("div#map_area>div#map_layers_manager>input[name='current_vectorlayer_fill']").val();
                var strokeColor = '#' +
                    $("div#map_area>div#map_layers_manager>input[name='current_vectorlayer_stroke']").val();

                var styleFunction = styleFunctionAutomat(layerIndex, false, fillColor, strokeColor, true);
                var Layer = new ol.layer.Vector({
                        source: new ol.source.Vector({}),
                        style: styleFunction
                });

                vectorLayers.push(Layer);
                map.addLayer(Layer);
                $("#map_layers_manager>select[name='current_vectorlayer']").append("<option value='" + layerIndex +
                    "'>" + layerNameComplete + "</option>");
                var currentLayerItem = $("#map_layers_vector_list>li#" + layerIndex + "vec");

                currentLayerItem.css('background-color', fillColor);
                currentLayerItem.css('color', strokeColor);

                //event Handler-Visibility
                currentLayerItem.children("input#map_layers_visibility_" + layerIndex).change(function(){
                        Layer.setVisible(!Layer.getVisible());
                });

                //event Handler-Labels
                currentLayerItem.children("select").change(function(){
                        var labels = true;
                        var localLayerIndex = $(this).val();
                        if($(this).children("option:selected").text() === "---"){ labels = false; }
                        var styleFunction = styleFunctionAutomat(localLayerIndex, labels);
                        Layer.setStyle(styleFunction);
                });

                vectorLayerCount++;

                //set Results-Add-to-Map Button visible
                if($("#query_results_container table tr.selected").length > 0){
                        $("#query_results_area>button:nth-child(3)").show();
                }

        }
});

//************************************************************************
//                      event-Handler for changing Style
//************************************************************************

$("#map_layers_manager>button[name='map_setStyle_button']").click(function(){
        var layerSelectItem =
           $("div#map_area>div#map_layers_manager>select[name='current_vectorlayer']>option:selected");
        var layerindex = layerSelectItem.val();

        if(layerindex){
                var layer = vectorLayers[layerindex];
                var fillColor = '#' +
                    $("div#map_area>div#map_layers_manager>input[name='current_vectorlayer_fill']").val();
                var strokeColor = '#' +
                    $("div#map_area>div#map_layers_manager>input[name='current_vectorlayer_stroke']").val();
                var styleFunction = styleFunctionAutomat(layerindex, false, fillColor, strokeColor, false);

                layer.setStyle(styleFunction);

                var currentLayerItem = $("#map_layers_vector_list>li:nth-child(" + (layerindex+1) + ")");
                currentLayerItem.css('background-color', fillColor);
                currentLayerItem.css('color', strokeColor);
        }
});

//************************************************************************
//                event-Handler for Removing all Layers
//************************************************************************

$("#map_layers_manager>button[name='map_reset_button']").click(function(){
        for(var i = 0; i<vectorLayerCount; i++){ map.removeLayer(vectorLayers[i]); }
        for(var i2 = 1; i2<basemapCount; i2++){ map.removeLayer(baseMapLayers[i2]); }
        $("#map_layers_vector_list>li").remove();
        $("#map_layers_raster_list>li:not(:first-child)").remove();
        $("#map_layers_manager>select[name='current_vectorlayer']>option").remove();
        vectorLayerCount = 0;
        basemapCount = 1;
```

```
        vectorLayers.length = 0;
        baseMapLayers.length = 1;
        vectorLayerStyles.length = 0;
        $("#query_results_area>button:nth-child(3)").hide();
});


//*****************************************************************************
//                event-Handler for Adding Vector data to Map
//*****************************************************************************

$("#query_results_area>button:nth-child(3)").click(function(){
        var geomInput = $("#query_results_container th input[type='radio']:checked");
        var checkedGeomTHIndex = geomInput.val();
        var geomColName = geomInput.siblings("label").text().replace(".", "\n");
        var geomType = $("#query_results_container th:has(input[type='radio']:checked)").attr('class');
        var idIndices = [];
        $("#query_results_container th
           input[type='checkbox']:checked").each(function(){ idIndices.push($(this).val()); });

        var layerIndex = $("#map_layers_manager>select[name='current_vectorlayer']>option:selected").val();

        if(layerIndex && checkedGeomTHIndex && geomType && idIndices.length>0){

                //scroll to Map-Area
                $("html, body").animate({
                        scrollTop : $("#map_area").offset().top },
                        1000
                );

                var layerSource = vectorLayers[layerIndex].getSource();
                var formatReader = geomType === 'gml_lit' ? new ol.format.GML(): new ol.format.WKT();
                var features = [];
                $("#query_results_container tr.selected>td:nth-child(" + checkedGeomTHIndex + ")").each(function(){
                        var currentRow = $(this);
                        var feature = formatReader.readFeatures(currentRow.text()).pop();
                        var idString = geomColName;
                        for(var i = 0; i < idIndices.length; i++){
                                idString += "\n" + currentRow.siblings("td:nth-child(" + idIndices[i] + ")").text();
                        }

                        feature.set('name', idString);//OR SET AS ID
                        features.push(feature);
                });
                layerSource.addFeatures(features);
                view.fit(layerSource.getExtent(), map.getSize());
        }
});

//*****************************************************************************
//                    Event-Handler for basemap-Visibility
//*****************************************************************************

$("div#map_layers_area>ul#map_layers_raster_list>li:first-child>input#map_basmaps_visibility_0").change(function(){
        baseMapLayers[0].setVisible(!baseMapLayers[0].getVisible());
});

//*****************************************************************************
//                              Event-Handler for WMS
//*****************************************************************************

$("div#wms_wfs_area>div#wms_area>button").click(function(){
        var button = $(this);
        var url = button.siblings("#wms_url").val();
        var layerName = button.siblings("#wms_layer").val();

        var layer = new ol.layer.Tile({
                source: new ol.source.TileWMS({
                        url: url,
                        params: { LAYERS: layerName },
                        hidpi: false
                })
        });

        map.addLayer(layer);

        var layerindex = basemapCount++;
        baseMapLayers[layerindex] = layer;

        var layerNameComplete = layerindex + "-" + layerName;

        $("div#map_layers_area>ul#map_layers_raster_list").append("<li>" +
                "<input type='checkbox' id='map_basmaps_visibility_" + layerindex + "' value='" + layerindex + "'
                     checked/>" + "<label>" + layerNameComplete + "</label>" + "</li>");

        $("div#map_layers_area>ul#map_layers_raster_list>li>input#map_basmaps_visibility_" +
           layerindex).change(function(){
                var basemapIndex = $(this).val();
                baseMapLayers[basemapIndex].setVisible(!baseMapLayers[basemapIndex].getVisible());
        });

        //scroll to Map-Area
        $("html, body").animate({
                scrollTop : $("#map_area").offset().top },
                1000
        );

});

//*****************************************************************************
//                              Event-Handler for WFS
//*****************************************************************************
```

```
	$("div#wms_wfs_area>div#wfs_area>button").click(function(){

		var url = $("#wms_wfs_area>div#wfs_area>input#wfs_url").val();
		var layerName = $("#wms_wfs_area>div#wfs_area>input#wfs_layer").val();
		var outputFormat = $("#wms_wfs_area>div#wfs_area>select>option:selected").val();

		var format;
		if(outputFormat === 'gml' || outputFormat === 'GML'){ format = new ol.format.GML(); }
		else if(outputFormat === 'json' || outputFormat === 'application/json'){ format = new ol.format.GeoJSON(); }
		else if(outputFormat === 'gml3' || outputFormat === 'GML3'){ format = new ol.format.GML3(); }
		else if(outputFormat === 'kml' || outputFormat === 'KML'){ format = new ol.format.KML(); }
		else{ return; }

		if(url && layerName){
			var requestString = url + /*"?" + */encodeURI("service=WFS&version=1.1.0&request=GetFeature&typename="
			    + layerName + "&outputFormat=" + outputFormat + "&srsname=EPSG:3857&bbox=");
			var layerIndex = vectorLayerCount;
			var layerNameComplete = (layerIndex + 1) + "-" + layerName;

			$("#map_layers_vector_list").append("<li id='" + layerIndex +
				"wfs'><input type='checkbox' id='map_layers_visibility_" + layerIndex + "' value='" +
				layerIndex + "' checked/><label>" + layerNameComplete + "</label></li>");

			var fillColor = '#' +
			    $("div#map_area>div#map_layers_manager>input[name='current_vectorlayer_fill']").val();
			var strokeColor = '#' +
			    $("div#map_area>div#map_layers_manager>input[name='current_vectorlayer_stroke']").val();
			var styleFunction = styleFunctionAutomat(layerIndex, false, fillColor, strokeColor, true);

			var Layer = new ol.layer.Vector({
				source: new ol.source.Vector({
					format: format,
					url: function(extent){
						var returnString = requestString + encodeURI(extent.join(',') + ",EPSG:3857");
					       return returnString;
					},
					strategy: ol.loadingstrategy.bbox
				}),
				style: styleFunction
			});

			vectorLayers.push(Layer);
			map.addLayer(Layer);

			$("#map_layers_manager>select[name='current_vectorlayer']").append("<option value='" + layerIndex +
			    "'>" + layerNameComplete + "</option>");

			var currentLayerItem = $("#map_layers_vector_list>li#" + layerIndex + "wfs");
			currentLayerItem.css('background-color', fillColor);
			currentLayerItem.css('color', strokeColor);

			//event Handler-Visibility
			currentLayerItem.children("input#map_layers_visibility_" + layerIndex).change(function(){
				Layer.setVisible(!Layer.getVisible());
			});

			vectorLayerCount++;

			//scroll to Map-Area
			$("html, body").animate({
			       scrollTop : $("#map_area").offset().top },
			       1000
			);
		}
	});
})();


//*****************************************************************************
//                          Download-Functions
//*****************************************************************************

(function(){
	$("#datastruct>ul#datastruct_tableslist>li.datastruct_table>button").click(function(){
		var datastruct = $("div#datastruct_name").text();
		var glossaryName = $(this).val();
		var requestText = "http://" + $(window.location).attr('hostname') + "/GetDoc.php?" +
			"requesttype=glossary&datastructure=" + datastruct + "&name=" + glossaryName;

		var newWindow = window.open(requestText, '_blank');
		if(newWindow){ newWindow.focus(); }
		else{ alert('Please activate Pop-ups'); }
	});

	$("div#datastruct_diagram>div#datastruct_panel button#download_datastructure_schema").click(function(){
		var requestText = "http://" + $(window.location).attr('hostname') + "/GetDoc.php?" +
			"requesttype=datastructureschema";

		var newWindow = window.open(requestText, '_blank');
		if(newWindow){ newWindow.focus(); }
		else{ alert('Please activate Pop-ups'); }
	});

	$("div#datastruct_diagram>div#datastruct_panel button#download_datastructure").click(function(){
		var datastruct = $("div#datastruct_name").text();
		var requestText = "http://" + $(window.location).attr('hostname') + "/GetDoc.php?" +
			"requesttype=datastructure&datastructure=" + datastruct;

		var newWindow = window.open(requestText, '_blank');
		if(newWindow){ newWindow.focus(); }
		else{ alert('Please activate Pop-ups'); }
	});
```

```
        $("div#datastruct diagram>div#datastruct panel button#download glossary schema").click(function(){
                var requestText = "http://" + $(window.location).attr('hostname') + "/GetDoc.php?" +
                        "requesttype=glossaryschema";

                var newWindow = window.open(requestText, '_blank');
                if(newWindow){ newWindow.focus(); }
                else{ alert('Please activate Pop-ups'); }
        });
})();


//*****************************************************************************
//                              3D-Functions
//*****************************************************************************

(function(){

        //get Container
        var container = $("#ThreeD_map_visualisation_area");
        var containerWidth = container.width();
        var containerHeight = container.height();

        //make scene
        var scene = new THREE.Scene();

        //make renderer
        var renderer = new THREE.WebGLRenderer();
        renderer.setClearColor(new THREE.Color(0xb3b3cc), 1);
        renderer.setSize(containerWidth, containerHeight);
        //renderer.shadowMapEnabled = true;

        //add Ambient-Light
        var ambientLight = new THREE.AmbientLight(new THREE.Color(0xCCDDFF));
        scene.add(ambientLight);

        //add Directional-Light
        var directionalLight = new THREE.DirectionalLight(new THREE.Color(0xFFFFFF), 0.8);
        directionalLight.position.x = 0;
        directionalLight.position.y = 0;
        directionalLight.position.z = 400;
        //directionalLight.castShadow = true;
        //directionalLight.shadowDarkness = 1;
        scene.add(directionalLight);

        //Normalizer-Values (to avoid large Coordinate-Values)
        var minX = 0;
        var maxX = 0;
        var minY = 0;
        var maxY = 0;
        var minZ = 0;
        var maxZ = 0;
        var xnormalizer = 0;
        var ynormalizer = 0;
                        var znormalizer = 0;
        var coordNormalizerSet =  false;

        //initial Position of Camera
        var camera = new THREE.PerspectiveCamera(45, containerWidth/containerHeight, 0.1, 100000);
        camera.position.x = 0;
        camera.position.y = 0;
        camera.position.z = 80;
        camera.lookAt(scene.position);

        //Container must me set BEFORE trachballControls
        container.append(renderer.domElement);

        //set GeometryGroups
        //Point-Group
        var points = new THREE.Group();
        var linestrings = new THREE.Group();
        var polygons = new THREE.Group();
        var surfaces = new THREE.Group();
        /*points.castShadow = true;
        points.receiveShadow = false;
        linestrings.castShadow = true;
        linestrings.receiveShadow = false;
        polygons.castShadow = true;
        polygons.receiveShadow = false;*/
        scene.add(points);
        scene.add(linestrings);
        scene.add(polygons);
        scene.add(surfaces);

        //Coordinates-Group
        var zline;
        var xline;
        var yline;

        //Ground-plane
        var plane = new THREE.Mesh(new THREE.PlaneGeometry(60,60,1,1), new THREE.MeshLambertMaterial({color: 0x4D4D4D}));
        plane.position.x = 0;
        plane.position.y = 0;
        plane.position.z = 0;
        /*plane.castShadow = false;
        plane.receiveShadow = true;*/
        scene.add(plane);

        //set Controls
        var trackballControls = new THREE.TrackballControls(camera, renderer.domElement);
        trackballControls.rotateSpeed = 5.0;
        trackballControls.zoomSpeed = 1.5;
        trackballControls.panSpeed = 1.0;
        var clock = new THREE.Clock();
```

```
function render() {
        var delta = clock.getDelta();
        trackballControls.update(delta);
        requestAnimationFrame(render);
        renderer.render(scene, camera);
}

/*renderer.*/render(/*scene, camera*/);


//*************************************************************************
//                            Add-Ground-Plane
//*************************************************************************

//Ground-plane
function addGround(){
        var deltaX = maxX-minX;
        var deltaY = maxY-minY;

        plane = new THREE.Mesh(new THREE.PlaneGeometry(deltaX + 3, deltaY + 3, 1, 1), new
           THREE.MeshLambertMaterial({color: 0x4D4D4D}));
        plane.position.x = minX + deltaX/2;
        plane.position.y = minY + deltaY/2;
        plane.position.z = minZ;
        /*plane.castShadow = false;
        plane.receiveShadow = true;*/

        if(zline){
                scene.remove(zline);
                scene.remove(xline);
                scene.remove(yline);
        }

        var zlineCurve = new THREE.Geometry();
        zlineCurve.vertices.push(new THREE.Vector3(minX-1.5, minY-1.5, minZ));
        zlineCurve.vertices.push(new THREE.Vector3(minX-1.5, minY-1.5, maxZ));
        var zlinematerial = new THREE.LineBasicMaterial({opacity: 1.0, linewidth: 1, color: 0x33CC33 });
        zline = new THREE.Line(zlineCurve, zlinematerial);
        scene.add(zline);

        var xlineCurve = new THREE.Geometry();
        xlineCurve.vertices.push(new THREE.Vector3(minX-1.5, minY-1.5, minZ));
        xlineCurve.vertices.push(new THREE.Vector3(maxX+3, minY-1.5, minZ));
        var xlinematerial = new THREE.LineBasicMaterial({opacity: 1.0, linewidth: 1, color: 0xff0000 });
        xline = new THREE.Line(xlineCurve, xlinematerial);
        scene.add(xline);

        var ylineCurve = new THREE.Geometry();
        ylineCurve.vertices.push(new THREE.Vector3(minX-1.5, minY-1.5, minZ));
        ylineCurve.vertices.push(new THREE.Vector3(minX-1.5, maxY+3, minZ));
        var ylinematerial = new THREE.LineBasicMaterial({opacity: 1.0, linewidth: 1, color: 0x0000ff });
        yline = new THREE.Line(ylineCurve, ylinematerial);
        scene.add(yline);

        scene.add(plane);
}

//*************************************************************************
//               functions to retrieve Geometries from WKT
//*************************************************************************

function get3DPointCoordsFromWKT(wellKnownText){
        if(wellKnownText.match(/^POINT Z\(-?\d+(\.\d+)?( -?\d+(\.\d+)?){2}\)$/) ||
           wellKnownText.match(/^POINT ZM\(-?\d+(\.\d+)?( -?\d+(\.\d+)?){3}\)$/)){
                var localString = wellKnownText.replace(/^POINT ZM?\(/, '').replace(')', '');
                var coords = localString.split(' ');

                if(coordNormalizerSet){
                        coords[0] = Number(xnormalizer) + Number(coords[0]);
                        coords[1] = Number(ynormalizer) + Number(coords[1]);
                        coords[2] = Number(znormalizer) + Number(coords[2]);
                }
                else{
                        xnormalizer = 0-Number(coords[0]);
                        ynormalizer = 0-Number(coords[1]);
                        znormalizer = 0-Number(coords[2]);
                        coords[0] = 0;
                        coords[1] = 0;
                        coords[2] = 0;
                        coordNormalizerSet = true;
                }

                minX = minX > coords[0]? coords[0] : minX;
                maxX = maxX < coords[0]? coords[0] : maxX;
                minY = minY > coords[1]? coords[1] : minY;
                maxY = maxY < coords[1]? coords[1] : maxY;
                minZ = minZ > coords[2]? coords[2] : minZ;
                maxZ = maxZ < coords[2]? coords[2] : maxZ;

                return coords;
        }
        else{ return false; }
}

function get3DLinestringFromWKT(wellKnownText){
        if(wellKnownText.match(/^LINESTRING Z\(-?\d+(\.\d+)?( -?\d+(\.\d+)?){2}(, -?\d+(\.\d+)?( -
           ?\d+(\.\d+)?){2}){1,}\)$/) ||
           wellKnownText.match(/^LINESTRING ZM\(-?\d+(\.\d+)?( -?\d+(\.\d+)?){3}(, -?\d+(\.\d+)?( -
           ?\d+(\.\d+)?){3}){1,}\)$/)){
                var localString = wellKnownText.replace(/^LINESTRING ZM?\(/, '').replace(')', '');
                var verticesArray = localString.split(', ');
                var verticesCount = verticesArray.length;
```

```
                var vertices = [];

                for(var i = 0; i < verticesCount; i++){
                        var coords = verticesArray[i].split(' ');

                        if(i === 0 && !coordNormalizerSet){
                                xnormalizer = 0-Number(coords[0]);
                                ynormalizer = 0-Number(coords[1]);
                                znormalizer = 0-Number(coords[2]);
                                coords[0] = 0;
                                coords[1] = 0;
                                coords[2] = 0;
                                coordNormalizerSet = true;
                        }
                        else{
                                coords[0] = Number(xnormalizer) + Number(coords[0]);
                                coords[1] = Number(ynormalizer) + Number(coords[1]);
                               coords[2] = Number(znormalizer) + Number(coords[2]);
                        }

                        minX = minX > coords[0]? coords[0] : minX;
                        maxX = maxX < coords[0]? coords[0] : maxX;
                        minY = minY > coords[1]? coords[1] : minY;
                        maxY = maxY < coords[1]? coords[1] : maxY;
                        minZ = minZ > coords[2]? coords[2] : minZ;
                        maxZ = maxZ < coords[2]? coords[2] : maxZ;

                        vertices[i] = new THREE.Vector3(Number(coords[0]), Number(coords[1]), Number(coords[2]));

                }
                return vertices;
        }
        else{ return false; }
}

function get3DPolygonFromWKT(wellKnownText){
        if(wellKnownText.match(/^POLYGON Z\(\(-?\d+(\.\d+)?( -?\d+(\.\d+)?){2}(, -?\d+(\.\d+)?( -
           ?\d+(\.\d+)?){2}){1,}\)(, \(-?\d+(\.\d+)?( -?\d+(\.\d+)?){2}(, -?\d+(\.\d+)?( -?\d+(\.\d+)?){2}){1,}
           \))*\)$/) || wellKnownText.match(/^POLYGON ZM\(\(-?\d+(\.\d+)?( -?\d+(\.\d+)?){3}(, -?\d+(\.\d+)?( -
           ?\d+(\.\d+)?){3}){1,}\)(, \(-?\d+(\.\d+)?( -?\d+(\.\d+)?){3}(, -?\d+(\.\d+)?( -?\d+(\.\d+)?){3})
           {1,}\))*\)$/)){
                var localString = wellKnownText.replace(/^POLYGON ZM?/, '').replace('((', '').replace('))', '');
                var polygons = localString.split('), (');
                var polygonCount = polygons.length;
                var polygonsArray = [];
                for(var i = 0; i < polygonCount; i++){
                        var verticesArray = polygons[i].split(', ');
                        var verticesCount = verticesArray.length;
                        var vertices = [];
                        for(var j = 0; j < verticesCount; j++){
                                var coords = verticesArray[j].split(' ');

                                if(i === 0 && j === 0 && !coordNormalizerSet){
                                        xnormalizer = 0-Number(coords[0]);
                                        ynormalizer = 0-Number(coords[1]);
                                        znormalizer = 0-Number(coords[2]);
                                        coords[0] = 0;
                                        coords[1] = 0;
                                        coords[2] = 0;
                                        coordNormalizerSet = true;
                                }
                                else{
                                        coords[0] = Number(xnormalizer) + Number(coords[0]);
                                        coords[1] = Number(ynormalizer) + Number(coords[1]);
                                        coords[2] = Number(znormalizer) + Number(coords[2]);
                                }

                                minX = minX > coords[0]? coords[0] : minX;
                                maxX = maxX < coords[0]? coords[0] : maxX;
                                minY = minY > coords[1]? coords[1] : minY;
                                maxY = maxY < coords[1]? coords[1] : maxY;
                                minZ = minZ > coords[2]? coords[2] : minZ;
                                maxZ = maxZ < coords[2]? coords[2] : maxZ;

                                vertices[j] = new THREE.Vector3(Number(coords[0]), Number(coords[1]),
                                      Number(coords[2]));
                        }
                        polygonsArray[i] = vertices;
                }
                return polygonsArray;
        }
        else{ return false; }
}


//*****************************************************************************
//                event-Handler for Adding Geometries to 3D-Map
//*****************************************************************************

$("#query_results_area>button:nth-child(4)").click(function(){
        var geomInput = $("#query_results_container th input[type='radio']:checked");
        var checkedGeomTHIndex = geomInput.val();
        var geomColName = geomInput.siblings("label").text();
        var geomType = $("#query_results_container th:has(input[type='radio']:checked)").attr('class');
        var idIndices = [];
        $("#query_results_container th
           input[type='checkbox']:checked").each(function(){ idIndices.push($(this).val()); });

        if(checkedGeomTHIndex && geomType ==='wkt_lit' && idIndices.length>0){

                //scroll to Map-Area
                $("html, body").animate({
                        scrollTop : $("#ThreeD_map_area").offset().top }, 1000 );
```

```
var currentColor = '#' + $("#ThreeD_map_area>#ThreeD_map_layers_manager input.jscolor").val();

$("#query_results_container tr.selected>td:nth-child(" + checkedGeomTHIndex + ")").each(function(){
        var currentRow = $(this);
        var wkt = currentRow.text();

        //get ID-String
        var idString = geomColName;
        for(var i = 0; i < idIndices.length; i++){
                idString += "\n" + currentRow.siblings("td:nth-child(" + idIndices[i] + ")").text();
        }

        if(wkt.startsWith('POINT Z')){

                var coords =  get3DPointCoordsFromWKT(wkt);
                if(coords !== false){
                        scene.remove(plane);
                        var geometry = new THREE.SphereGeometry(0.1, 20, 20);
                        var sphereMaterial = new THREE.MeshLambertMaterial({color: currentColor });
                        var sphere = new THREE.Mesh(geometry,sphereMaterial);
                        sphere.position.x = coords[0];
                        sphere.position.y = coords[1];
                        sphere.position.z = coords[2];
                        sphere.currentScale = 1;
                        //sphere.castShadow = true;
                        var pointIndex = points.children.length;
                        points.add(sphere);
                        $("#ThreeD_map_geometrieslist").append("<li>" + idString + "</li>");
                        $("#ThreeD_map_area>#ThreeD_map_layers_manager>select").append("<option
                                value='POINT" + pointIndex + "'>" + idString + "</option>");
                        $("#ThreeD_map_geometrieslist>li:last-child").css('color', currentColor);
                        $("#ThreeD_map_layers_manager>select>option:last-child").css('background-
                                color', currentColor);
                }
        }

        else if(wkt.match(/^MULTIPOINT Z\(-?\d+(\.\d+)?( -?\d+(\.\d+)?){2}(, -?\d+(\.\d+)?( -
              ?\d+(\.\d+)?){2})*\)$/) ||
                 wkt.match(/^MULTIPOINT ZM\(-?\d+(\.\d+)?( -?\d+(\.\d+)?){3}(, -?\d+(\.\d+)?( -
                        ?\d+(\.\d+)?){3})*\)$/)){

                var localString = wkt.replace(/^MULTIPOINT ZM?\(/, '').replace(/\(/g,
                        '').replace(/\)/g, '');
                var multipointsTuples = localString.split(', ');
                var multipointsCount = multipointsTuples.length;

                var multipoints = new THREE.Group();

                for(var m = 0; m < multipointsCount; m++){
                        var multipointCoords =  get3DPointCoordsFromWKT((wkt.startsWith('POINT ZM')?
                       'POINT ZM(' : 'POINT Z(') + multipointsTuples[m] + ')');
                        if(multipointCoords !== false){
                                scene.remove(plane);
                                var multip_geometry = new THREE.SphereGeometry(0.1, 20, 20);
                                var multip_sphereMaterial = new THREE.MeshLambertMaterial({color:
                                        currentColor });
                                var multip_sphere = new THREE.Mesh(multip_geometry,
                                        multip_sphereMaterial);
                                multip_sphere.position.x = multipointCoords[0];
                                multip_sphere.position.y = multipointCoords[1];
                                multip_sphere.position.z = multipointCoords[2];
                                //multip_sphere.castShadow = true;
                                multipoints.add(multip_sphere);
                                multip_sphere.currentScale = 1;
                        }
                }
                var multipointIndex = points.children.length;
                points.add(multipoints);
                $("#ThreeD_map_geometrieslist").append("<li>" + idString + "</li>");
                $("#ThreeD_map_area>#ThreeD_map_layers_manager>select").append("<option value='POINT"
                      + multipointIndex + "'>" + idString + "</option>");
                $("#ThreeD_map_geometrieslist>li:last-child").css('color', currentColor);
                $("#ThreeD_map_layers_manager>select>option:last-child").css('background-color',
                       currentColor);
        }

        else if(wkt.startsWith('LINESTRING Z')){
                var vertices = get3DLinestringFromWKT(wkt);
                if(vertices !== false){
                        scene.remove(plane);
                        var lines = new THREE.SplineCurve3(vertices);
                        var lineTubeGeometry = new THREE.TubeGeometry(lines, vertices.length+2, 0.1,
                                12, false);
                        var linematerial = new THREE.MeshLambertMaterial({color: currentColor });
                        var line = new THREE.Mesh(lineTubeGeometry, linematerial);
                        line.originalPath = lines;
                        line.oldSegmentsCount = vertices.length+2;
                        line.oldRadius = 0.1;
                        line.originalMaterial = linematerial;
                        var lineIndex = linestrings.children.length;
                        linestrings.add(line);

                        $("#ThreeD_map_geometrieslist").append("<li>" + idString + "</li>");
                        $("#ThreeD_map_area>#ThreeD_map_layers_manager>select").append("<option
                                value='LINE" + lineIndex + "'>" + idString + "</option>");
                        $("#ThreeD_map_geometrieslist>li:last-child").css('color', currentColor);
                        $("#ThreeD_map_layers_manager>select>option:last-child").css('background-
                                color', currentColor);
                }
        }
```

```
else if(wkt.match(/^MULTILINESTRING Z\(\(-?\d+(\.\d+)?( -?\d+(\.\d+)?){2}(, -?\d+(\.\d+)?( -
      ?\d+(\.\d+)?){2}){1,}\)(, \(-?\d+(\.\d+)?( -?\d+(\.\d+)?){2}(, -?\d+(\.\d+)?( -?\d+
      (\.\d+)?){2}){1,}\))*\)$/) || wkt.match(/^MULTILINESTRING ZM\(\(-?\d+(\.\d+)?( -
      ?\d+(\.\d+)?){3}(, -?\d+(\.\d+)?( -?\d+(\.\d+)?){3}){1,}\)(, \(-?\d+(\.\d+)?( -?\d+
      (\.\d+)?){3}(, -?\d+(\.\d+)?( -?\d+(\.\d+)?){3}){1,}\))*\)$/)){

        scene.remove(plane);

        var multilline_localString = wkt.replace(/^MULTILINESTRING ZM?\(/, '');
        var multilline_Tuples = multilline_localString.split('), (');
        var multilline_Count = multilline_Tuples.length;

        var multilines = new THREE.Group();

        for(var n = 0; n < multilline_Count; n++){
                var multilineVertices =  get3DLinestringFromWKT((wkt.startsWith('LINESTRING
                        ZM')? 'LINESTRING ZM(' : 'LINESTRING Z(') + multilline_Tuples[n].
                       replace(/\(/g, '').replace(/\)/g, '') + ')');
                if(multilineVertices !== false){
                        var curve = new THREE.SplineCurve3(multilineVertices);
                        var multilineTubeGeometry = new THREE.TubeGeometry(curve,
                                multilineVertices.length*2, 0.1, 12, false);
                        var multilineMaterial = new THREE.MeshLambertMaterial({color:
                                currentColor });
                        var multilineline = new THREE.Mesh(multilineTubeGeometry,
                                multilineMaterial);
                        multilineline.originalPath = curve;
                        multilineline.oldSegmentsCount = multilineVertices.length+2;
                        multilineline.oldRadius = 0.1;
                        multilineline.originalMaterial = multilineMaterial;
                        multilines.add(multilineline);
                }
        }

        var multilineIndex = linestrings.children.length;
        linestrings.add(multilines);
        $("#ThreeD_map_geometrieslist").append("<li>" + idString + "</li>");
        $("#ThreeD_map_area>#ThreeD_map_layers_manager>select").append("<option value='LINE" +
              multilineIndex + "'>" + idString + "</option>");
        $("#ThreeD_map_geometrieslist>li:last-child").css('color', currentColor);
        $("#ThreeD_map_layers_manager>select>option:last-child").css('background-color',
              currentColor);
}

else if(wkt.startsWith('POLYGON Z')){
        var verticesBundle = get3DPolygonFromWKT(wkt);
        if(verticesBundle !== false){
                scene.remove(plane);
                var polygonCount = verticesBundle.length;
                var polygonAll = new THREE.Group();

                for(var j = 0; j < polygonCount; j++){
                        var polygongeom = new THREE.SplineCurve3(verticesBundle[j]);
                        var polygonTubeGeometry = new THREE.TubeGeometry(polygongeom,
                                verticesBundle[j].length*2, 0.1, 12, true);
                        var polygonMaterial = new THREE.MeshLambertMaterial({color:
                                currentColor });
                        var polygon = new THREE.Mesh(polygonTubeGeometry, polygonMaterial);
                        polygon.originalPath = polygongeom;
                        polygon.oldSegmentsCount = verticesBundle[j].length*2;
                        polygon.oldRadius = 0.1;
                        polygon.originalMaterial = polygonMaterial;
                        polygonAll.add(polygon);
                }

                var polygonIndex = polygons.children.length;
                polygons.add(polygonAll);
                $("#ThreeD_map_geometrieslist").append("<li>" + idString + "</li>");
                $("#ThreeD_map_area>#ThreeD_map_layers_manager>select").append("<option
                        value='POLYGON" + polygonIndex + "'>" + idString + "</option>");
                $("#ThreeD map geometrieslist>li:last-child").css('color', currentColor);
                $("#ThreeD_map_layers_manager>select>option:last-child").css('background-
                        color', currentColor);
        }
}

//check for triangulated-Surface
else if(wkt.match(/^MULTIPOLYGON Z\(\(\(-?\d+(\.\d+)?( -?\d+(\.\d+)?){2}(, -?\d+(\.\d+)?( -
      ?\d+(\.\d+)?){2}){2}\)){2,}\)\)$/)){

        scene.remove(plane);

        var surface_localString = wkt.replace(/^MULTIPOLYGON ZM?\(/, '');
        var surface_Tuples = surface_localString.split(')), ((');
        var surface_Count = surface_Tuples.length;
        var surface = new THREE.Geometry();

        for(var q = 0; q < surface_Count; q++){
                var triangleWKT = "";

                if(q === 0){ triangleWKT = 'MULTIPOLYGON Z' + surface_Tuples[q].replace('(((',
                        '((') + '))'; }
                else if(q === (multipoly_Count-1)){ triangleWKT = 'MULTIPOLYGON Z' + '((' +
                        surface_Tuples[q].replace(')))', '))'); }
                else{ triangleWKT = 'MULTIPOLYGON Z' + '((' + surface_Tuples[q] + '))'; }

                var surface_verticesBundle = get3DPolygonFromWKT(triangleWKT);

                if(surface_verticesBundle !== false){
                        surface.vertices.push(surface_verticesBundle[0][0]);
                        surface.vertices.push(surface_verticesBundle[0][1]);
                        surface.vertices.push(surface_verticesBundle[0][2]);
                        var coordCounter = q*3;
```

```
                                        surface.faces.push(coordCounter, coordCounter++, coordCounter++);
                                }
                        }

                        var surfaceMaterial = new THREE.MeshLambertMaterial({color: currentColor });
                        var surfaceMesh = new THREE.Mesh(surface, surfaceMaterial);

                        var surfaceIndex = surfaces.children.length;
                        surfaces.add(surfaceMesh);
                        $("#ThreeD_map_geometrieslist").append("<li>" + idString + "</li>");
                        $("#ThreeD_map_area>#ThreeD_map_layers_manager>select").append("<option
                              value='SURFACE" + surfaceIndex + "'>" + idString + "</option>");
                        $("#ThreeD_map_geometrieslist>li:last-child").css('color', currentColor);
                        $("#ThreeD_map_layers_manager>select>option:last-child").css('background-color',
                              currentColor);
                }

                else if(wkt.match(/^MULTIPOLYGON Z\(\(\(-?\d+(\.\d+)?( -?\d+(\.\d+)?){2}(, -?\d+(\.\d+)?( -
                      ?\d+(\.\d+)?){2}){1,}\)(, \(-?\d+(\.\d+)?( -?\d+(\.\d+)?){2}(, -?\d+(\.\d+)?( -?\d+
                      (\.\d+)?){2}){1,}\))*\)(, \(\(-?\d+(\.\d+)?( -?\d+(\.\d+)?){2}(, -?\d+(\.\d+)?( -
                      ?\d+(\.\d+)?){2}){1,}\)(, \(-?\d+(\.\d+)?( -?\d+(\.\d+)?){2}(, -?\d+(\.\d+)?( -
                      ?\d+(\.\d+)?){2}){1,}\))*\))+\)$/) ||
                         wkt.match(/^MULTIPOLYGON ZM\(\(\(-?\d+(\.\d+)?( -?\d+(\.\d+)?){3}(, -?\d+(\.\d+)?( -
                      ?\d+(\.\d+)?){3}){1,}\)(, \(-?\d+(\.\d+)?( -?\d+(\.\d+)?){3}(, -?\d+(\.\d+)?( -?\d+
                      (\.\d+)?){3}){1,}\))*\)(, \(\(-?\d+(\.\d+)?( -?\d+(\.\d+)?){3}(, -?\d+(\.\d+)?( -?\d+
                      (\.\d+)?){3}){1,}\)(, \(-?\d+(\.\d+)?( -?\d+(\.\d+)?){3}(, -?\d+(\.\d+)?
                      ( -?\d+(\.\d+)?){3}){1,}\))*\))+\)$/)){

                        scene.remove(plane);

                        var multipoly_localString = wkt.replace(/^MULTIPOLYGON ZM?\(/, '');
                        var multipoly_Tuples = multipoly_localString.split(')), ((');
                        var multipoly_Count = multipoly_Tuples.length;

                        var multipolygon = new THREE.Group();

                        var dimension = wkt.startsWith('POLYGON ZM')? 'POLYGON ZM' : 'POLYGON Z';

                        for(var o = 0; o < multipoly_Count; o++){
                                var polygonWKT = "";
                                if(o === 0){
                                        polygonWKT = dimension + multipoly_Tuples[o].replace('(((', '((') +
                                        '))';
                                }
                                else if(o === (multipoly_Count-1)){
                                        polygonWKT = dimension + '((' + multipoly_Tuples[o].replace(')))',
                                        '))');
                                }
                                else{
                                        polygonWKT = dimension + '((' + multipoly_Tuples[o] + '))';
                                }

                                var multipoly_verticesBundle = get3DPolygonFromWKT(polygonWKT);

                                if(multipoly_verticesBundle !== false){

                                        var multipoly_polygonCount = multipoly_verticesBundle.length;
                                        var multipoly_polygonAll = new THREE.Group();

                                        for(var p = 0; p < multipoly_polygonCount; p++){
                                                var multipolygeom = new
                                                 THREE.SplineCurve3(multipoly_verticesBundle[p]);
                                                var multipolyTubeGeometry = new
                                                 THREE.TubeGeometry(multipolygeom,
                                                multipoly_verticesBundle[p].length*2, 0.1, 12, true);
                                                var multipolyMaterial = new THREE.MeshLambertMaterial({color:
                                                 currentColor });
                                                var multipolypolygon = new THREE.Mesh(multipolyTubeGeometry,
                                                 multipolyMaterial);
                                                multipolypolygon.originalPath = multipolygeom;
                                                multipolypolygon.oldSegmentsCount =
                                                 multipoly_verticesBundle[p].length*2;
                                                multipolypolygon.oldRadius = 0.1;
                                                multipolypolygon.originalMaterial = multipolyMaterial;
                                                multipoly_polygonAll.add(multipolypolygon);
                                        }

                                        multipolygon.add(multipoly_polygonAll);

                                }
                        }

                        var multipoly_Index = polygons.children.length;
                        polygons.add(multipolygon);
                        $("#ThreeD map geometrieslist").append("<li>" + idString + "</li>");
                        $("#ThreeD_map_area>#ThreeD_map_layers_manager>select").append("<option
                              value='POLYGON" + multipoly_Index + "'>" + idString + "</option>");
                        $("#ThreeD_map_geometrieslist>li:last-child").css('color', currentColor);
                        $("#ThreeD_map_layers_manager>select>option:last-child").css('background-color',
                              currentColor);
                }

        });//End-looping through Result-Rows

        //position Light
        directionalLight.position.z = maxZ + 10;

        //position Ground
        addGround();
    }
});//End Event-Handler for Adding Geometry to 3D-Scene
```

```
//*****************************************************************************
//                        event-Handler to reset 3D-Map
//*****************************************************************************

$("div#ThreeD_map_area>div#ThreeD_map_layers_manager>button:nth-child(5)").click(function(){
        scene.remove(plane);

        if(zline){
                scene.remove(zline);
                scene.remove(xline);
                scene.remove(yline);
        }

        plane = new THREE.Mesh(new THREE.PlaneGeometry(60,60,1,1), new THREE.MeshLambertMaterial({color: 0x4D4D4D}));
        plane.position.x = 0;
        plane.position.y = 0;
        plane.position.z = 0;
        /*plane.castShadow = false;
        plane.receiveShadow = true;*/
        scene.add(plane);

        scene.remove(points);
        points = new THREE.Group();
        scene.remove(linestrings);
        linestrings = new THREE.Group();
        scene.remove(polygons);
        polygons = new THREE.Group();
        scene.remove(surfaces);
        surfaces = new THREE.Group();

        scene.add(points);
        scene.add(linestrings);
        scene.add(polygons);
        scene.add(surfaces);

        /*points.castShadow = true;
        points.receiveShadow = false;
        linestrings.castShadow = true;
        linestrings.receiveShadow = false;
        polygons.castShadow = true;
        polygons.receiveShadow = false;*/

        camera.position.x = 0;
        camera.position.y = 0;
        camera.position.z = 80;
        trackballControls.target.set(0, 0, 0);

        minX = 0;
        maxX = 0;
        minY = 0;
        maxY = 0;
        minZ = 0;
        maxZ = 0;
        xnormalizer = 0;
        ynormalizer = 0;
                                        znormalizer = 0;
        coordNormalizerSet =  false;


        $("#ThreeD_map_geometrieslist>li").remove();
        $("#ThreeD_map_layers_manager>select>option").remove();

});

//*****************************************************************************
//                      event-Handler to zoom to Geometry
//*****************************************************************************

$("div#ThreeD_map_area>div#ThreeD_map_layers_manager>button:nth-child(4)").click(function(){
        var currentGeomName = $(this).siblings('select').children('option:checked').val();
        var object;

        if(!currentGeomName){ object = plane; }
        else if(currentGeomName.startsWith('POINT')){
                var pointIndex = currentGeomName.replace('POINT', '');
                object = points.children[pointIndex];
        }
        else if(currentGeomName.startsWith('LINE')){
                var lineIndex = currentGeomName.replace('LINE', '');
                object = linestrings.children[lineIndex];
        }
        else if(currentGeomName.startsWith('POLYGON')){
                var polygonIndex = currentGeomName.replace('POLYGON', '');
                object = polygons.children[polygonIndex];
        }
        else if(currentGeomName.startsWith('SURFACE')){
                var surfaceIndex = currentGeomName.replace('SURFACE', '');
                object = surfaces.children[surfaceIndex];
        }

        var bbox = new THREE.Box3().setFromObject(object);
        var position = bbox.getCenter();
        var size = bbox.getSize();

        camera.position.x = position.x;
        camera.position.y = position.y;
        camera.position.z = position.z + (size.x>size.y? 1.5*size.x : 1.5*size.y);

        trackballControls.target.set(position.x, position.y, position.z);

});
```

```
	//*****************************************************************************
	//                    event-Handler to resize Geometry
	//*****************************************************************************

	function resizeTube(parent, childIndex, up, closeGeom){
		var tube = parent.children[childIndex];
		if(tube instanceof THREE.Mesh){
			var lineTubeGeometry = new THREE.TubeGeometry(
				tube.originalPath,
				tube.oldSegmentsCount,
				(up? tube.oldRadius*2 : tube.oldRadius/2), 12,
				closeGeom
			);
			var newTube = new THREE.Mesh(lineTubeGeometry, tube.originalMaterial);
			newTube.originalPath = tube.originalPath;
			newTube.oldSegmentsCount = tube.oldSegmentsCount;
			newTube.oldRadius = up? tube.oldRadius*2 : tube.oldRadius/2;
			newTube.originalMaterial = tube.originalMaterial;
			parent.children[childIndex] = newTube;
		}
		else{
			var childCount = tube.children.length;
			for(var i = 0; i < childCount; i++){ resizeTube(tube, i, up, closeGeom); }
		}
	}

	function resizeGeometries(currentGeomName, up){
		if(!currentGeomName){ return; }
		else if(currentGeomName.startsWith('POINT')){
			var pointIndex = currentGeomName.replace('POINT', '');
			var object = points.children[pointIndex];
			if(object instanceof THREE.Mesh){
				var newScale = up? object.currentScale*1.5 : object.currentScale/1.5;
				object.currentScale = newScale;
				object.scale.x = newScale;
				object.scale.y = newScale;
				object.scale.z = newScale;
			}
			else{
				var itemsCount = object.children.length;
				for(var i = 0; i < itemsCount; i++){
					var newPointScale = up? object.children[i].currentScale*1.5 :
					 object.children[i].currentScale/1.5;
					object.children[i].currentScale = newPointScale;
					object.children[i].scale.x = newPointScale;
					object.children[i].scale.y = newPointScale;
					object.children[i].scale.z = newPointScale;
				}
			}
		}
		else{
			var collection;
			var closeGeom;
			var geomIndex;

			if(currentGeomName.startsWith('LINE')){
				geomIndex = currentGeomName.replace('LINE', '');
				collection = linestrings;
				closeGeom = false;
			}
			else if(currentGeomName.startsWith('POLYGON')){
				geomIndex = currentGeomName.replace('POLYGON', '');
				collection = polygons;
				closeGeom = true;
			}

			resizeTube(collection, geomIndex, up, closeGeom);

		}
	}


	$("div#ThreeD_map_area>div#ThreeD_map_layers_manager>button:nth-child(6)").click(function(){
		var currentGeomName = $(this).siblings('select').children('option:checked').val();
		resizeGeometries(currentGeomName, true);
	});

	$("div#ThreeD_map_area>div#ThreeD_map_layers_manager>button:nth-child(7)").click(function(){
		var currentGeomName = $(this).siblings('select').children('option:checked').val();
		resizeGeometries(currentGeomName, false);
	});

})();


//*****************************************************************************
//                              Download-Function
//*****************************************************************************

(function(){

	$('#binary_download>button').click(function(){
		var datastructure = $('#datastruct_name').text();
		var repository = $("#repositories_list input[name='Repository']:checked");
		if(!repository.length){ return; }
		var repositoryName = repository.val();
		var fileName = $('#binary_download>input').val().trim();
		var fileParameter = "";
		if(fileName && fileName != 'Filename.suffix'){
			if(!fileName.match(/^\w+.\w+$/)){ return; }
			else{ fileParameter = '&filename=' + fileName; }
		}
```

```
                window.open("http://" + $(window.location).attr('hostname') + "/GetFile.php?datastructure=" + datastructure +
                        "&repositoryname=" + repositoryName + fileParameter, '_blank');
        });

        $('#binary_download>input').on('input', function(){
                if($('#binary_download>input').val().trim()){
                        $('#binary_download>button').text('get File');
                }
                else{ $('#binary_download>button').text('get Repository'); }
        });

    })();

});
```

GETDOC.PHP

```php
//*****************************************************************************
//Written by Thomas Winklehner
//on 01.11.2016
//Purpose:
//        - facilitate documents' download
//*****************************************************************************

<?php

    if ($_SERVER['REQUEST_METHOD'] === 'GET'){
        if(isset($_GET['requesttype'])){
            $requestType = $_GET['requesttype'];
            $HomeDir = realpath(__DIR__);
            for($count = 0; $count < 2; $count++){
                $HomeDir = dirname($HomeDir);
            };

            //Datastructure-Schema requested
            if($requestType === 'datastructureschema'){
                $path = $HomeDir . DIRECTORY_SEPARATOR . 'XMLDataBaseStructure' .
                    DIRECTORY_SEPARATOR . 'DatabaseStructureSchema' . '.xsd';
                $file = fopen($path, 'r') or die('Cannot open file');
                header('Content-Type: text/xml');
                echo fread($file, filesize($path));
                fclose($file);
            }

            //Datastructure requested
            else if($requestType === 'datastructure'){
                if(isset($_GET['datastructure'])){
                    $name = $_GET['datastructure'];
                    if(preg_match("/^[\w]+$/", $name)){
                        $path = $HomeDir . DIRECTORY_SEPARATOR . 'DatabaseStructures' .
                                DIRECTORY_SEPARATOR . $name . '.xml';
                        $file = fopen($path, 'r') or die('Cannot open file');
                        header('Content-Type: text/xml');
                        echo fread($file, filesize($path));
                        fclose($file);
                    }
                    else{ echo "Invalid name-parameter"; }
                }
                else{ echo "Missing datastructure-parameter"; }
            }

            //Glossaryschema requested
            else if($requestType === 'glossaryschema'){
                $path = $HomeDir . DIRECTORY_SEPARATOR . 'GlossarySchema' .
                        DIRECTORY_SEPARATOR . 'GlossarySchema' . '.xsd';
                $file = fopen($path, 'r') or die('Cannot open file');
                header('Content-Type: text/xml');
                echo fread($file, filesize($path));
                fclose($file);
            }

            //Glossary requested
            else if($requestType === 'glossary'){
                if(isset($_GET['name']) && isset($_GET['datastructure'])){
                    $datastructure = $_GET['datastructure'];
                    $name = $_GET['name'];
                    if(preg_match("/^[\w]+$/", $name) && preg_match("/^[\w]+$/", $datastructure)){
                        $path = $HomeDir . DIRECTORY_SEPARATOR . 'Glossaries' .
                                DIRECTORY_SEPARATOR . $datastructure . DIRECTORY_SEPARATOR . $name . '.xml';
                        $file = fopen($path, 'r') or die('Cannot open file');
                        header('Content-Type: text/xml');
                        echo fread($file, filesize($path));
                        fclose($file);
                    }
                    else{ echo "Invalid parameter"; }
                }
                else{ echo "Missing parameter"; }
            }
            else{ echo "Missing parameter"; }
        }
        else{ echo "Missing requesttype-parameter"; }
    }
    else{ echo "False Request Method: '" . $_SERVER['REQUEST_METHOD'] . "'. 'GET' required"; };
?>
```

QUERY.PHP

```
//************************************************************************
//Written by Thomas Winklehner
//on 02.11.2016
//Purpose:
//       - query repositories
//************************************************************************
<?php

    if ($_SERVER['REQUEST_METHOD'] == 'POST'){

        //************************************************************************
        //                              Constants
        //************************************************************************

            $MainConnectionUtilityPath = realpath(__DIR__);
            for($count = 0; $count < 2; $count++){
                $MainConnectionUtilityPath = dirname($MainConnectionUtilityPath);
            };
            $HomeDir = $MainConnectionUtilityPath;
            $ScripsPath = $MainConnectionUtilityPath . DIRECTORY_SEPARATOR . 'Scripts' . DIRECTORY_SEPARATOR;
            $MainConnectionUtilityPath = $ScripsPath . 'MainConnectionUtility.php';

        //************************************************************************
        //                          Including Scripts
        //************************************************************************

            if(file_exists($MainConnectionUtilityPath)){
                include_once $MainConnectionUtilityPath; }
            else{
                echo "<p>Script 'MainConnectionUtility.php' was not found!";
                return;
            }

        //************************************************************************
        //                          Aggregate-Values
        //************************************************************************

            $Aggregates = array(
                1=>'count(*)',
                2=>'count',
                3=>'avg',
                4=>'max',
                5=>'min',
                6=>'sum',
                7=>'total'
            );

        //************************************************************************
        //                          Geometry-Names
        //************************************************************************

            $GeometryNames = array(
                1=>'POINT',
                2=>'LINESTRING',
                3=>'POLYGON',
                4=>'MULTIPOINT',
                5=>'MULTILINESTRING',
                6=>'MULTIPOLYGON'
            );

        //************************************************************************
        //                      Query Values from Table
        //************************************************************************

            //check Existence of Repository-Name
            if(!isset($_POST['datastruct'])){  echo '<p>' . htmlentities($RETURNVALUES[6], ENT_QUOTES) . '</p>'; }
            else if(!isset($_POST['repository'])){ echo '<p>' . htmlentities($RETURNVALUES[5], ENT_QUOTES) . '</p>'; return; }
            else if(!isset($_POST['fromString'])){ echo '<p>' . htmlentities($RETURNVALUES[7], ENT_QUOTES) . '</p>'; return; };

        //++++++++++++++++++++++++++++++++++++++++++++++++++Table- and ColumnNames-Arrays

            $TABLESinQuery = array();
            $COLUMNSinQuery = array();

        //++++++++++++++++++++++++++++++++++++++++++++++++++prepare From-String

            //get From-String and check
            $fromStringOrig = str_replace(" ", "", $_POST['fromString']);
            if(!preg_match("/^[\w]+(;[\w]+\([\w]+\.[\w]+=[\w]+\.[\w]+\))*$/", $fromStringOrig)){ echo '<p>' .
         $RETURNVALUES[8] . '</p>'; return; };

            //get first Table
            $fromArray = explode(';', $fromStringOrig);
            $tableCount = sizeof($fromArray);

            //SQL-From-String
```

```
    $sqlFromString = " FROM " . $fromArray[0];
    $TABLESinQuery[] = $fromArray[0];//store Table in Tables-Array

    //loop through the other Tables and append Statements to From-String
    for($i= 1; $i < $tableCount; $i++){
        $tableString = $fromArray[$i];
        $tableValues = explode('(', $tableString);
        $tablename = $tableValues[0];
        $joinedColumns = explode(')', $tableValues[1])[0];
        $TABLESinQuery[] = $tablename;//store Table in Tables-Array
        $sqlFromString .= " JOIN " . $tablename . " ON (" . $tableValues[1];
        $singleJoinedColumns = explode('=', $joinedColumns);
        $COLUMNSinQuery[] = $singleJoinedColumns[0];//store Column in Columns-Array
        $COLUMNSinQuery[] = $singleJoinedColumns[1];//store Column in Columns-Array
    };

//++++++++++++++++++++++++++++++++++++++++++++++++++prepare Select-Columns

    $spatialColumns = array();
    $spatialColumnFound = false;

    $sqlSelectString = "SELECT ";
    $aggregateSet = false;
    if(!isset($_POST['selectedColumnsString'])){ echo "<p>" . $RETURNVALUES[9] . "</p>"; return; }
    else{
        $selectStringOrig = str_replace(" ", "", $_POST['selectedColumnsString']);
        if(!preg_match("/^[\w]+\.[\w]+(;[\w]+\.[\w]+)*$/", $selectStringOrig)){
            echo "<p>" . $RETURNVALUES[10] . "</p>"; return;
        }
        else{
            $aggregateString = str_replace(" ", "", $_POST['aggregate']);
            if(empty($aggregateString) || !in_array($aggregateString, $Aggregates)){
                $selectArray = $fromArray = explode(';', $selectStringOrig);
                $columnName = $selectArray[0];
                $sqlSelectString .= $columnName;
                $COLUMNSinQuery[] = $columnName;//store Column in Columns-Array
                $columnCount = sizeof($selectArray);
                if(stripos($columnName, '_GML_') !== false){
                   $spatialColumns[] = " class='gml_lit'";
                   $spatialColumnFound = true; }
                else if(stripos($columnName, '_WKT_') !== false){
                    $spatialColumns[] = " class='wkt_lit'"; $spatialColumnFound = true; }
                else{ $spatialColumns[] = "";};

                for($i= 1; $i < $columnCount; $i++){
                    $otherColName = $selectArray[$i];
                    $COLUMNSinQuery[] = $otherColName;//store Column in Columns-Array
                    if(stripos($otherColName, '_GML_') !== false){
                       $spatialColumns[] = " class='gml_lit'";
                       $spatialColumnFound = true; }
                    else if(stripos($otherColName, '_WKT_') !== false){
                       $spatialColumns[] = " class='wkt_lit'"; $spatialColumnFound = true; }
                    else{ $spatialColumns[] = "";};
                    $sqlSelectString .= ", " . $selectArray[$i];
                };
            }
            else{
                $firstColumn = explode(';', $selectStringOrig)[0];
                $COLUMNSinQuery[] = $firstColumn;//store Column in Columns-Array
                $sqlSelectString .= $aggregateString . "(" . $firstColumn . ")"; $aggregateSet = true;
            };
        };
    };

//++++++++++++++++++++++++++++++++++++++++++++++++++prepare GroupBy-String

    $sqlGroupByString = "";
    if($aggregateSet){
        if(isset($_POST['groupByString'])){
            $groupByStringOrig = str_replace(" ", "", $_POST['groupByString']);
            if(preg_match("/^[\w]+\.[\w]+(;[\w]+\.[\w]+)*$/", $groupByStringOrig)){
                $groupByValues = explode(';', $groupByStringOrig);
                $sqlGroupByString .= " GROUP BY " . $groupByValues[0];
                $COLUMNSinQuery[] = $groupByValues[0];//store Column in Columns-Array
                $groupByCount = sizeof($groupByValues);
                for($i= 1; $i < $groupByCount; $i++){
                    $COLUMNSinQuery[] = $groupByValues[$i];//store Column in Columns-Array
                    $sqlGroupByString .= ", " . $groupByValues[$i];
                };
            };
        };
    };

//++++++++++++++++++++++++++++++++++++++++++++++++++prepare OrderBy-String

    $sqlOrderByString = "";
    if(isset($_POST['orderString'])){
        $oderByStringOrig = str_replace(" ", "", $_POST['orderString']);
        if(preg_match("/^[\w]+\.[\w]+;(ASC|DESC)(;[\w]+\.[\w]+;(ASC|DESC))*$/", $oderByStringOrig)){
```

```
            $oderByValues = explode(';', $oderByStringOrig);
            $sqlOrderByString .= " ORDER BY " . $oderByValues[0] . " " . $oderByValues[1];
            $COLUMNSinQuery[] = $oderByValues[0];//store Column in Columns-Array
            $oderByCount = sizeof($oderByValues);
            for($i=2; $i < $oderByCount; $i=$i+2){
                $COLUMNSinQuery[] = $oderByValues[$i];//store Column in Columns-Array
                $sqlOrderByString .= ", " . $oderByValues[$i] . " " . $oderByValues[$i+1];
            };
        };
    };

//++++++++++++++++++++++++++++++++++++++++++++++++++prepare conditions-String

    $sqlWhereString = "";
    if(isset($_POST['conditionsString'])){
        $conditionsStringOrig = str_replace(" ", "", $_POST['conditionsString']);
        if(preg_match("/^[\w]+\.[\w]+;(([<>=!]{1,2};[[:digit:]]+(\.[[:digit:]]+)*)|((=|!=|LIKE);'[^']*'))(;[\w]+\.[\w]
          +;(([<>=!]{1,2};[[:digit:]]+(\.[[:digit:]]+)*)|((=|!=|LIKE);'[^']*')))*$/", $conditionsStringOrig)){
            $conditionsValues = explode(';', $conditionsStringOrig);
            $operator = $conditionsValues[1];
            if($operator === "LIKE"){ $operator = " " . $operator . " "; };
            $sqlWhereString .= " WHERE " . $conditionsValues[0] . $operator . $conditionsValues[2];
            $COLUMNSinQuery[] = $conditionsValues[0];//store Column in Columns-Array
            $conditionsCount = sizeof($conditionsValues);
            for($i= 3; $i < $conditionsCount; $i = $i+3){
                $COLUMNSinQuery[] = $conditionsValues[$i];//store Column in Columns-Array
                $localOperator = $conditionsValues[$i+1];
                if($localOperator === "LIKE"){ $localOperator = " " . $localOperator . " "; };
                $sqlWhereString .= " AND " . $conditionsValues[$i] . $localOperator . $conditionsValues[$i+2];
            };
        };
    };

    $limit = " LIMIT ";
    if(isset($_POST['sqlLimit']) && preg_match("/^[[:digit:]]{1,6}$/", $_POST['sqlLimit'])){ $limit .= $_POST['sqlLimit']; }
    else{ $limit .= '1000'; }

    $offset = " OFFSET ";
    if(isset($_POST['sqlOffset']) && preg_match("/^[[:digit:]]{1,6}$/", $_POST['sqlOffset'])){ $offset .=
          $_POST['sqlOffset']; }
    else{ $offset .= '0'; }

    $sql = $sqlSelectString . $sqlFromString . $sqlWhereString . $sqlGroupByString . $sqlOrderByString . $limit . $offset;

    echo "<p>Query: " . $sql . "<p>";


//++++++++++++++++++++++++++++++++++++++++++++++++++start Connection

//*************************************************************************
//                          getting RepositoryPath
//*************************************************************************

    $datastructure = str_replace(" ", "", $_POST['datastruct']);
    $datastructure = str_replace("'", "", $datastructure);

    $repository = str_replace(" ", "", $_POST['repository']);
    $repository = str_replace("'", "", $repository);

    $repositoryPath = getRepositoryPath($datastructure, $repository);

    if(is_integer($repositoryPath)){
        echo "<p>" . $RETURNVALUES[$repositoryPath] . "</p>";
        return;
    }
    else if(!file_exists($repositoryPath)){
        echo "<p>Repository " . $repository . " not found!</p>";
        return;
    };

//*************************************************************************
//                          Establish Repository-Connection
//*************************************************************************

    $dbHandle = new SQLiteDBConn($repositoryPath);
    if(isset($dbHandle)){

        $TableColumns = array();

        //check Columns
        $TABLESinQueryCount = sizeof($TABLESinQuery);
        for($i = 0; $i < $TABLESinQueryCount; $i++){
            $TableColumnsResult = $dbHandle->query("Pragma table_info(" . $TABLESinQuery[$i] . ")");
            $row = $TableColumnsResult->fetchArray(SQLITE3_ASSOC);
            while($row !== false){
                $TableColumns[$TABLESinQuery[$i] . "." . $row['name']] = $row['type'];
                $row = $TableColumnsResult->fetchArray(SQLITE3_ASSOC);
            };
            $TableColumnsResult->finalize();
```

```
            };
            $COLUMNSinQueryCount = sizeof($COLUMNSinQuery);

            for($i = 0; $i < $COLUMNSinQueryCount; $i++){
                $ColumnName = $COLUMNSinQuery[$i];
                if(!array_key_exists($ColumnName, $TableColumns)){
                    echo "<p>Column " . $ColumnName . " does not exist!</p>";
                    return;
                };
                $ColumnType = $TableColumns[$ColumnName];
                if(stripos($ColumnType, 'BLOB') === false && !in_array($ColumnType, $GeometryNames)){ continue; }
                else{
                    echo "<p>Forbidden Column-Type " . $ColumnType . " of Column " . $COLUMNSinQuery[$i] . "!</p>";
                    return;
                };
            };

            //Result
            $Result = $dbHandle->query($sql);

            //get Header
            $columnsCount = $Result->numColumns();
            $tableHeader = "<table><tr><th>row</th>";
            $headerValues = explode(';', $_POST['selectedColumnsString']);
            $headerValuesCount = sizeof($headerValues);
            $checked = false;
            for($i=0; $i < $headerValuesCount;$i++){
                if($spatialColumnFound){
                    if(empty($spatialColumns[$i])){
                       $tableHeader .= "<th><input type='checkbox' value='" . ($i+2) . "' checked/><label>" .
                       htmlentities($headerValues[$i], ENT_QUOTES) . "</label></th>"; }

                    //++++++++++++++++++++++EXCLUDE GML from being chosen as Geometry
                    else if($spatialColumns[$i] === " class='gml_lit'"){
                        $tableHeader .= "<th>" . htmlentities($headerValues[$i], ENT_QUOTES) . "</th>";
                    }
                    //++++++++++++++++++++++EXCLUDE GML from being chosen as Geometry

                    else{
                        $checkedString = "";
                        if(!$checked){
                            $tableHeader .= "<th " . $spatialColumns[$i] . "><input type='radio'
                                name='query_results_spatial_cols_radio' value='" .
                                        ($i+2) . "' checked/><label>" .
                                htmlentities($headerValues[$i], ENT_QUOTES) . "</label></th>";
                            $checked = true;
                        }
                        else{
                            $tableHeader .= "<th " . $spatialColumns[$i] . "><input type='radio'
                                name='query_results_spatial_cols_radio' value='" .
                                        ($i+2) . "'/><label>" .
                                htmlentities($headerValues[$i], ENT_QUOTES) . "</label></th>";
                        };
                    };
                }
                else{ $tableHeader .= "<th>" . htmlentities($headerValues[$i], ENT_QUOTES) . "</th>"; }
            };

            $tableHeader .= "</tr>";
            echo $tableHeader;

            //get Result-Values
            $counter = 0;
            $row = $Result->fetchArray(SQLITE3_NUM);
            while($row !== false){
                $tableRow;
                if( $spatialColumnFound){ $tableRow = "<tr class='unselected'><td>" . ++$counter . "</td>"; }
                else{ $tableRow = "<tr><td>" . ++$counter . "</td>"; }
                for($i=0; $i < $columnsCount;$i++){
                    $tableRow .= "<td" . $spatialColumns[$i] . ">" . htmlentities($row[$i], ENT_QUOTES) . "</td>";
                };
                $tableRow .= "</tr>";
                echo $tableRow;
                $row = $Result->fetchArray(SQLITE3_NUM);
            };
            $Result->finalize();
            echo "</table>";
        }
        else{
            echo "<p>" . $RETURNVALUES[4] . "</p>";
            return;
        };

    }
    else{
        echo "Invalid Request Type. Use: POST";
    }
?>
```

GETFILE.PHP

```php
//*****************************************************************************
//Written by Thomas Winklehner
//on 03.11.2016
//Purpose:
//        - facilitate files' download
//*****************************************************************************
<?php
    if ($_SERVER['REQUEST_METHOD'] === 'GET'){
        if(isset($_GET['datastructure']) && isset($_GET['repositoryname'])){

            $HomeDir = realpath(__DIR__);
            for($count = 0; $count < 2; $count++){
                $HomeDir = dirname($HomeDir);
            };

            //get Parameters
            $datastructure = $_GET['datastructure'];
            $repositoryname = $_GET['repositoryname'];
            $fileName = "";

            //check Parameters
            if(!preg_match("/^[\w]+$/", $datastructure)){
                echo "invalid datastructure-Parameter";
                return; }
            else if(!preg_match("/^[\w]+$/", $repositoryname)){
                echo "invalid repository-Parameter";
                return;
            };

            //get File-Directory
            $filepath = $HomeDir . DIRECTORY_SEPARATOR . 'Repositories' . DIRECTORY_SEPARATOR . $datastructure . DIRECTORY_SEPARATOR .
                  $repositoryname;

            //get Filename
            if(isset($_GET['filename'])){
                $fileName = $_GET['filename'];
                $filepath .= DIRECTORY_SEPARATOR . $fileName; }
            else{
                $fileName = $repositoryname . '.sqlite';
                $filepath .= '.sqlite';
            };

            //check-FileName
            if(!preg_match("/^[\w]+.[\w]+$/", $datastructure)){
                echo "invalid datastructure-Parameter";
                return;
            }

            if(file_exists($filepath) && is_file($filepath)){

                //get file-size
                $size = filesize($filepath);

                if($size > 10000000000){
                    echo "File too big. Max size: 9 GB";
                    return;
                };

                //no time-out for script
                set_time_limit(0);

                //set Headers
                header('Content-Type: application/octet-stream');
                header('Content-Length: ' . $size);
                header('Content-Disposition: attachment; filename="' . $fileName . '"');
                header('Content-Transfer-Encoding: binary');
                header('Content-Description: File Transfer');

                //stop buffering
                ob_clean();
                ob_implicit_flush(false);
                ob_end_clean();
                readfile($filepath);

            }
            else{ echo "Download-failed. File not found or not-readable"; };
        }
        else{ echo "Download-failed. Missing valid Parameters"; };
    };
?>
```