

White Paper:
Revitalizing and Enhancing the Open Source 3D WebGIS of MayaArch3D
NEH Digital Humanities Advancement Grant, Level 1 (HAA-269061-20)

Heather Richards-Rissetto, School of Global Integrative Studies (SGIS), University of
Nebraska-Lincoln

Karin Dalziel, Center for Digital Research in the Humanities (CDRH), University of
Nebraska-Lincoln

Greg Tunink, Center for Digital Research in the Humanities (CDRH), University of
Nebraska-Lincoln

Michael Auer, Heidelberg Institute for Geoinformation Technology

Lukas Loos, Heidelberg Institute for Geoinformation Technology

PART I

Introduction

In archaeology, and more broadly cultural heritage, we tend to develop digital tools for specific projects with limited potential for customization and extensibility. This needs to change. We must develop software and infrastructure that is generalizable, well documented, and easier to maintain to facilitate cross-disciplinary scholarship in the Humanities. In the Digital Humanities, where development of customizable open source software is becoming common, there is still a void in the Spatial Humanities. For web mapping, many projects depend on proprietary software such as ArcGIS. Other projects employ open source options such as GeoServer, MapServer, and Leaflet; however, they do not have as complete a feature set as ESRI products and are not as fully integrated, requiring more technician time for development, and they do not offer 3D visualization and analysis. While ESRI offers 3D analysis desktop products, their 3D WebGIS web products are expensive and only offer 3D visualization, not online, real-time analysis.

Revitalizing goal

It is under these circumstances that the Phase I goal of “Revitalizing and Enhancing the open source 3D WebGIS of the MayaArch3D Project” was to evaluate the 3D WebGIS

software components in order to develop a plan to bring the system back online in a more extensible, better documented, more accessible, and easier to maintain form. Our objectives focused on evaluating current W3C open web standards, assessing the original system and code to identify necessary changes, and make recommendations for future implementation. A key find of our research is that to ensure successful implementation not only does the software require updates, the system requires improved data access and findability for georeferenced 3D data—a topic that is an ongoing challenge in cultural heritage and other fields.

History of MayaArch3D (2009-2015)

MayaArch3D is an innovative virtual research environment (VRE) that combines aspects of 2D, 3D, GIS, and archeological data into a platform that enables previously unavailable research queries, analysis, and 3D data visualization in real time on the web (Figure 1). From 2009-2015, the MayaArch3D software was developed and ran on servers at the Bruno Kessler Foundation (FBK) in Italy (prototype 2010-2013) and then at the German Archaeological Institute (DAI) in Germany when it started to degrade in 2017 and parts of the software stopped functioning (von Schwerin et al. 2009; von Schwerin et al. 2011; Reindel et al. 2016).

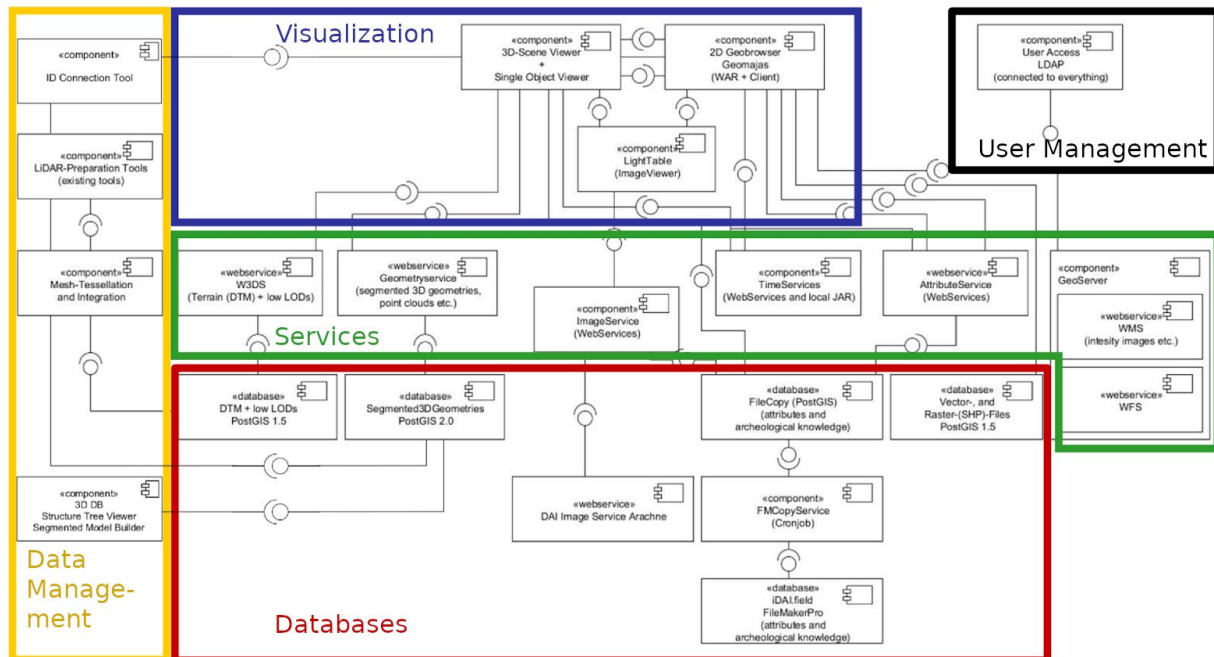


Fig. 1: Component diagram of MayaArch3D phase II.
Source: Translated from MayaArch3D Developer-Doc

In 2018, the Center for Digital Research in the Humanities (CDRH) and the DAI reached an agreement to transfer the project and its components to the University of Nebraska at Lincoln (UNL). Programmers at the CDRH looked at the files determining that we cannot simply install the system on a new server, and as currently configured, the software is not entirely open source, as it requires a proprietary license for VMware virtual server use. Moreover, we determined that a single database approach is not ideal; instead, building on our research from the NEH-funded [Keeping Data Alive Project](#), we realized that using an open source repository alongside other databases would better facilitate 3D data preservation, access, and reuse for the humanities. Despite these challenges, the 3D WebGIS is still cutting-edge—it is the only open source 3D WebGIS capable of 3D visibility analysis.

This level 1 Digital Humanities Advancement project—funded by NEH—evaluated the software components in order to develop recommendations to bring the 3D WebGIS software back online in a more extensible, better documented, more accessible, and easier to maintain form. Our strategy was to identify as much of the original MayaArch3D code base as possible, documenting it and compartmentalizing it so it is more extendable and reusable. We contend that a key component to interoperability, extensibility, and data reuse is implementing W3C open standards. While the original 3D WebGIS is based on open source programming languages and libraries including JavaScript, HTML, three.js, GeoJSON, and WebGL, W3C open web standards have continued to emerge since the system's original development (2013-2015). In addition to presenting a review of the original MayaArch3D system, we situate our recommendations within a discussion of new and emerging standards. Furthermore, based on our research we have divided future research and development into two main categories: (1) develop technical and socio-economic infrastructure to enable researchers to find, share, and analyze georeferenced 3D data on Maya archaeology (Wright and Richards 2019) and (2) provide software that researchers can customize to their own research areas (e.g., South America, Asia, etc.). Achievement of these two goals would allow researchers to carry out cross-disciplinary scholarship in spatial humanities (and beyond) and develop new 3D web-based computationally based methods.

Background

3D Virtual Research Environments

Increasingly, researchers are employing georeferenced 2D and 3D data to carry out spatial humanities and deep mapping research (Bodenhamer et al. 2010, 2015, Gregory et al. 2014). Typically these georeferenced data are project-specific, stored on hard drives, desktops, or local servers. Various humanities projects have created web mapping applications visualizing

georeferenced 2D data (typically georeferenced maps) and a few have developed online applications enabling real-time spatial analysis (e.g., [ORBIS](#), [CORONA ATLAS](#)). A shortcoming, and admittedly not a goal, of these projects is that researchers cannot easily (or independently) ingest their own data into the systems for comparative visualization and analysis. Moreover, and of relevancy to this project, is that an open source 3D WebGIS does not exist (with the exception of the original MayaArch3D) that allows researchers to ingest data to carry out real-time comparative 3D spatial analysis, and even the MayaArch3D system has several shortcomings and obstacles as a Virtual Research Environment (VRE)—that is, an online environment fostering collaborative research for georeferenced 3D data visualization and analysis (Kuroczynski 2017). While some shortcomings stemmed from technical challenges, one major reason for the system’s shortcomings (to facilitate 3D collaborative research) was socio-economic—we tried to research and develop too many components rather than focusing on a subset of components. [This topic will be discussed in more detail below.]

In archaeology and cultural heritage, and the humanities more broadly, researchers often work alone or in small, isolated groups, and we do not have an efficient way to carry out collaborative and comparative research, particularly across multiple projects. Thus, researchers may duplicate data and even infrastructure increasing costs, limiting opportunities, and decreasing efficiency. However, web-based solutions for virtual research environments offer a common workspace for projects, paving the way for interdisciplinary and comparative dialog. More importantly (for this project), data scattered around the world can be made accessible to scientists across the globe allowing for more diverse and comprehensive input and analysis. For cultural heritage in Europe, [Europeana](#) provides a web-based platform for data aggregation and access to over 52 million digital/digitized objects with metadata. While the platform provides some 3D objects such as those contributed by [3D-Icons](#), the system is not designed to deal with georeferenced 3D data and has only limited display capabilities.

Archaeological data are heterogeneous due to a variety of data types, diverse projects, legacy data, and more, making it difficult to bring together for comprehensive and comparative analysis, particularly in digital environments that require specific data structuring to ensure data interoperability (von Schwerin 2017). [Open Context](#)—a web research and publication platform, primarily focused on archaeology, works with researchers to structure data for digital publication for comparative analysis, and uses linked open standards to allow data findability (Kansa et al. 2020). For some projects, Open Context hosts georeferenced 2D data (i.e., kml) and increasingly works with researchers to host 3D objects (as OBJs). For example, a recent publication “Visualizing Votive Practice: Exploring Limestone and Terracotta

Sculpture from Athienou-Malloura through 3D Models” links to 3D models and associated data published on Open Context and viewable on Sketchfab (Counts et al. 2020). However, these are not georeferenced 3D models or scenes (i.e., 3D landscape models comprising multiple 3D models) and they are not downloadable for reuse on other platforms.

Another aspect of data heterogeneity that is challenging for 3D data (models and scenes) is a lack of standards. While archaeologists and other cultural heritage fields often use OBJs or COLLADA (.DAE) for single mesh 3D models or LAS files for 3D point clouds, which creates greater interoperability, data standards are less established or missing in 3D WebGIS. Thus, 3D WebGIS data is often incompatible making data integration and reuse time-consuming and challenging, if not impossible. The original MayaArch3D Project sought to address this issue, and while successful for a time, open standards for 3D WebGIS have changed since 2015, requiring a review of the current state of the system’s software. [A more detailed discussion is below.]

Beyond technical challenges, socio-economic circumstances are critical for successful VREs. The original MayaArch3D project primarily focused on innovative technological aspects (as was the purview of the funding) leading to challenges to offer a sustainable solution for long-term maintenance. Scholarly work is complex and multi-layered comprising: organizational settings, knowledge production, communication, and knowledge distribution. While a VRE can support each of these spheres, it is critical to identify what content and functionality should be available via the web vs. local solutions and when proprietary vs. open source solutions are better options, or whether a combination of solutions is needed. The next section provides a brief overview of the MayaArch3D System, its components and functionality, and its content (data and formats) as well as explanations of original choices in design and data formats to set the stage for our software review and recommendations.

Part II

Review of Technical Components, Software, and Data & Data Format of MayaArch3D System

Technical Components

The goal of Phase II of the MayaArch3D Project (2012-2015) was to develop a new research tool for archeology and monument conservation that brought together 3D technologies and GIS functionalities (2D and 3D) together in one internet platform for documentation and analysis of archaeological sites. The project developed concepts

and technical solutions for temporal, spatial, and attributive queries and were carried out iteratively and in collaboration between archaeologists, 3D metrologists, and computer scientists (specializing in geoinformatics) (Billen et al. 2013; Richards-Rissetto and von Schwerin 2017; von Schwerin et al. 2016; von Schwerin et al. 2017). The MayaArch3D system represents a web-based solution for the acquisition, management, analysis, and visualization of archaeological 2D and 3D geodata. The system comprises five components (Figure 1):

- Data Management
- Services
- Visualization
- Databases
- User Management

As part of the project, a database and server structure for server-side storage and processing of 3D, 2D and the attributive data were developed. Two key challenges were to integrate the existing archaeological databases of the DAI and to design and implement a suitable data model for the processing of georeferenced 3D models. In addition to the server-side components, web-based visualization and analysis tools were developed in a web platform. Archaeological data collected in the field and from archival resources are accessible online and complex queries and analyzes can also be done. In this area, a big challenge was the web based provision of heterogeneous data - archaeological (textual, numerical, etc.), georeferenced 2D, 2.5D, 3D data and models, and hierarchically segmented 3D models (von Schwerin et al. 2017).

Wherever it was possible, we employed already existing standardized open source solutions. Nevertheless, it was necessary to develop prototype solutions for some parts of the system when no standards were available. The 3DSceneViewer remained online through 2017 and the 3D Single Object Viewer functioned until 2019. Both systems eventually went offline because customized developments were not further developed or maintained and in some cases software dependencies changed.

Software

As part of the MayaArch3D project, many components were developed that are not necessary for handling the 3D data. These were developed to create a complete virtual research environment that provides the tools for working with 2D, 3D, and temporal data. All components are mentioned below and their functionality is briefly explained in order to understand the embedding of the 3D components in the system architecture. Each

component is explained from two perspectives: what a “User” gains from a component and what it contributes to the system from the perspective of a “Developer/Operator”. Note- We repeat Figure 1 below as it is color-coded to Table.

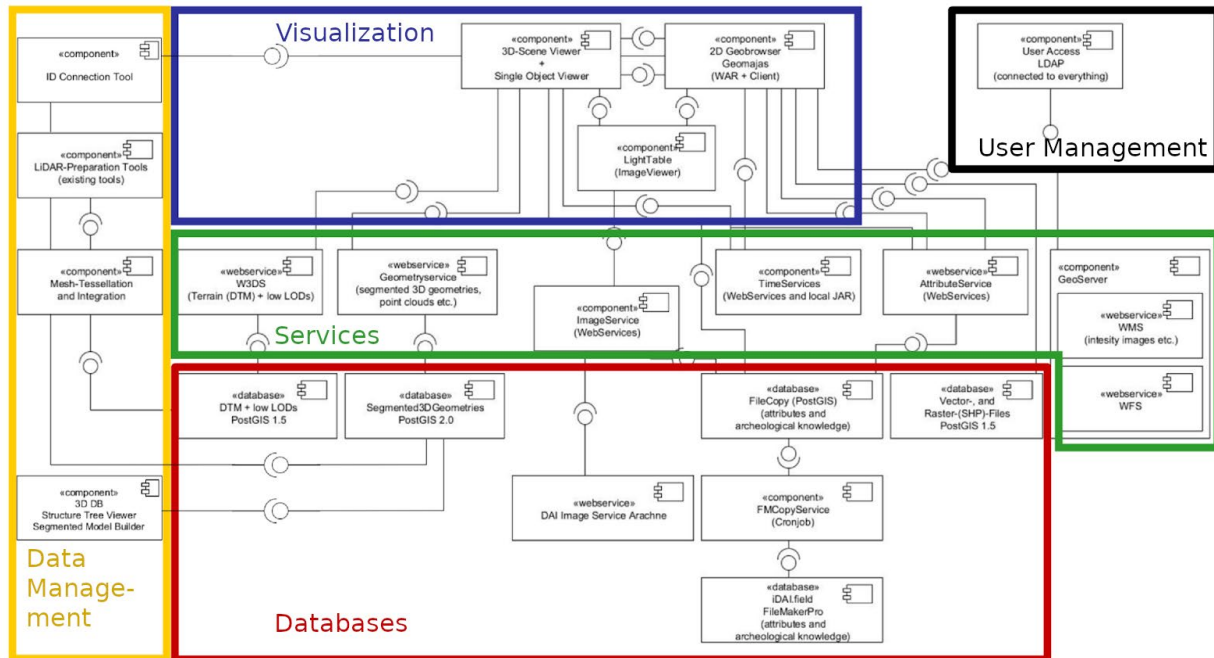


Fig. 1: Component diagram of MayaArch3D phase II. Translated from MayaArch3D-II Developer-Doc

Note: The following table (Tables 1-5) uses the corresponding colors of Fig. 1 to illustrate links between components. We have also included additional notes with explanations and recommendations for specific components.

Visualization:

Component name	Function
3D SceneViewer	<p>A web application to visualize and analyze large georeferenced 3D scenes. A custom implementation by GIScience Heidelberg and a project result of MayaArch3D-II.</p> <p>User:</p> <ul style="list-style-type: none"> ● Experience large scenes (that do not fit into RAM at once) in a seamless way (e.g. valley overview). ● Perform 3D spatial analysis on landscape and site scale (e.g. visibility analysis) ● Combine datasets by spatial location (georeference). 3D

	<p>Meshdata + 2D WebMapService Overlays from raster or vector sources</p> <ul style="list-style-type: none"> ● Overlay local user-defined datasets, without being forced to upload research data to a server <p>DevOp:</p> <ul style="list-style-type: none"> ● Single-Page website that was integrated in the MayaArch3D-WebGIS, but could be deployed on its own. ● Consumed data from W3DS, Geoserver and AttributeService ● Depends only on open-source libraries (GIScene.js using three.js, jQuery) <p>Note: This is the actual 3D WebGIS component. See also notes on W3DS for alternatives, e.g. Cesium</p>
3DSingleObjectViewer	<p>A web application to visualize single or segmented 3D models with or without georeferenced coordinates. Loads models from a service or file. A custom implementation by GIScience Heidelberg and a result of MayaArch3D-II.</p> <p>User:</p> <ul style="list-style-type: none"> ● View single or segmented models with high(er) resolution with or without georeferenced coordinates ● Use 3D model segments as interactive links to get information from databases (e.g. attributes from iDAIField) <p>DevOp:</p> <ul style="list-style-type: none"> ● Integrated into project website mayaarch3d.org to display (segmented) models from files (three.js SceneFormat - json) ● Consumed data from files served by an Apache Webservice on mayaarch3d.org, GeometryService, AttributeService ● Linked using Id Connection Tool
2DGeobrowser Geomajas	<p>A web application to visualize 2D Data for Central America and Copa based on Geomajas, an open source platform for Web GIS.</p> <p>User:</p> <ul style="list-style-type: none"> ● Explore locations of ancient Maya sites w/basic attributes ● Gateway to 3DSceneViewer
Light Table (Image viewer)	<p>A web application to show images from Arachne Object database of the DAI.</p> <p>DevOp:</p> <ul style="list-style-type: none"> ● Needs an ID connection.

Table 1: Visualization Components

Services:

Component name	Function
<p>W3DS</p> <p>Used for: 3D</p>	<p>W3DS stands for Web3DService. During the project (2013-2015) it was a custom Java-based implementation by GIScience Heidelberg based on a OGC W3DS discussion paper. At that time an official W3DS standard did not exist. In Oct 2017, an official OGC W3DS Standard was approved called OGC 3DP (3D Portrayal Service).</p> <p>User:</p> <ul style="list-style-type: none"> ● See 3D SceneViewer <p>DevOp:</p> <ul style="list-style-type: none"> ● Web service that consumes PostgreSQL/PostGIS tables (see DTM + low LODs) with georeferenced geometries and textured 3D model representations (VRML / three.js JSON) ● Delivered tiled data to the 3D SceneViewer ● Custom Java implementation of GIScience Heidelberg, extended during MayaArch3D-II to deliver three.js-JSON instead of VRML
<p>GeometryService</p> <p>Used for: 3D</p>	<p>The geometry service is used to dynamically build a hierarchical three.js-SCENE file (ver 3.2) from entries in the maya3d DB.</p>
<p>AttributeService</p> <p>Used for: 2D + 3D</p>	<p>Retrieves attributes information associated with archaeological structures. The geometry of the structures is stored in the maya3d DB while the attributes are maintained in a separate database at the DAI (iDAI.Field). Both databases are interconnected by an ID (see ID connection tool).</p> <p>DevOp:</p> <ul style="list-style-type: none"> ● JAVA ● As WAR-Archive in a Tomcat 7 servlet container <p>Note:</p> <p>The general problem of linking attributes to 2D/3D data was the potential risk of inconsistencies. Inconsistency means one database is changed by the user (e.g. the id) and then the link is broken. Why did we split Attributes and Geometry? Because iDAI.field already existed but was not capable of dealing with geometry data.</p> <p>Future: Best case is to avoid splitting the storage of geometry and attributes and keep them together as it is common for 2D geodata.</p> <p>Further discussion needed: What are alternative options to store and</p>

	<p>access 3D in a similar way as common for 2D but with extended options for more complexity and flexibility in terms of attribute structure (combination of object relational database management system and document store)? Short answer: Best to avoid the necessity of an isolated AttributeService in favor of an integrated FeatureInfoService</p>
ImageService	<p>Handles requests from Light Table and provides an API for getting Thumbnails, etc. The ImageService then requests the Arachne DB.</p> <p>DevOp:</p> <ul style="list-style-type: none"> ● JAVA ● As WAR-Archive in a Tomcat 7 servlet container
GeoServer	<p>Provides the 2D data for the Maya sites in Central America and the raster data derived from the lidar survey for copan for 2D Geobrowser.</p> <p>Provided image overlay data for 3D terrain tiles in 3D SceneViewer (lidar intensity, height color map).</p> <p>Note: The Geoserver is open source, easy to use, and implements many standards. We suggest that it be a component in any 2D and 3D geodata infrastructure. The key requirement was that the 3DSceneViewer was able to understand the standards provided by Geoserver, which means that we could have overlaid WMS-layers from any Geoserver provider in the world to combine with our 3D models on the fly. This component does not necessarily need to be maintained in the same system but could be loosely integrated.</p>

Table 2: Services Components

Databases:

Component name	Function
<p>DTM + low LODs</p> <p>Used for: 3D</p>	<p>User:</p> <ul style="list-style-type: none"> ● See 3D SceneViewer <p>DevOp:</p> <ul style="list-style-type: none"> ● Datasource served through W3DS to 3D-SceneViewer ● Streaming of large area datasets (Terrain, LOD 1 structures) using tiling and level-of-detail strategies ● Requires terrain to be preprocessed (tiling, multiple resolutions) with Mesh Tessellation and Integration

	<p>Note: At the end of MayaArch3D-II this database contained the whole Copan Terrain model as 3D-Tiles in 4 or 5 resolutions and a complete LOD 1 model of built structures (based on an extruded version of a 2D footprint vector dataset). These datasets could be used for web visualization and web-based visibility analysis. Thus, this database proved to be central for the usage, but was based mainly on non-standard pre-processing tools.</p>
<p>Segmented 3DGeometries</p> <p>Used for: 3D</p>	<p>User:</p> <ul style="list-style-type: none"> ● See 3D SingleObjectViewer <p>DevOp:</p> <ul style="list-style-type: none"> ● Datasource served through GeometryService to the 3D SingleObjectViewer ● Model segments with unique ids offer possibilities to link to further contextual information (e.g. archaeological attributes, images, etc.) from AttributeService and ImageService <p>Note: At the end of MayaArch3D-II this database only contained unsegmented LOD 1 building models. Reason: The process of creating semantically segmented models was very labor intensive and time consuming and had to be performed manually by 3D modeling experts.</p>
<p>FileCopy (PostGIS) attributes and archaeological knowledge</p>	<p>Database containing a copy of the Filemaker database in order to remedy shortcomings of the Filemaker jdbc driver.</p>
<p>iDAI.field FileMakerPro attributes</p>	<p>Archaeological database in Filemaker Pro that contains attributes and metadata for all digital objects (2D and 3D)</p>
<p>Vector- and Raster- (SHP)-Files PostGIS 1.5</p>	<p>Database for 2D data for Maya sites in Central America, 2D vector data for Copan (footprints, sculpture fall plans)</p> <p>User:</p> <ul style="list-style-type: none"> ● Links shapefiles to attribute database for spatial queries <p>DevOp:</p> <ul style="list-style-type: none"> ● Datasource Served through Geoserver to 3DSceneViewer and 2D Geobrowser Geomajas

Table 3: Databases Components

Data management:

Component name	Function
3D DB Structure Tree Viewer Segmented Model Builder	<p>Establishes hierarchical relationships of the segmented 3D models.</p> <p>User: A web application that allowed the user to inspect the semantic hierarchy of object types and how they were defined in the Ontology for Maya Structures. Uses structure tree to dynamically compose a hierarchical model scenefile by uploading single segment model files to the Ontology nodes. The result was visualized in the 3SingleObjectViewer</p> <p>DevOp: This was a html frontend using php to retrieve the ontology structure from the Segmented 3DGeometries database. The model composition was made client side using javascript and a hierarchy tree visualization using d3.js</p> <p>Note: This was used to create the hierarchical segmented model of Temple 18 reconstruction that is visible in the 3DObjectViewer on the mayaarch3d.org website.</p>
Mesh Tessellation and Integration	<p>Pre-processing tool to cut meshes of different LoDs into tiles to be served by W3DS.</p> <p>DevOp: The pipeline was:</p> <ol style="list-style-type: none">1. Import cleaned and classified point cloud of terrain into CloudCompare (see Lidar preparation tools)2. Convert point cloud into 2.5D DEMs

	<p>(GeoTIFF) with different resolutions</p> <ol style="list-style-type: none"> 3. Convert these different resolutions back to 3D meshes 4. Use this MeshTesselationTool to cut large meshes into tiles and store these into different database tables as datastore for the W3DS <p>Note: This tool was a custom implementation from GIScience Heidelberg during the MayaArch3D-II to be able to output the 3Dtiles in three.js-SceneFormat JSON. This is a Java implementation.</p>
Lidar preparation tools	Existing open source software: CloudCompare, MeshLab
ID connection tool	<p>Connect IDs of 3D objects with associated attributes in iDAI.field. “This ‘id connection’ is the key to create attribute queries on the iDAI.Field database triggered by 3D objects from the Postgres DB (see Dev Doc 1.2.2.5.1).</p> <p>Note: In MayaArch3D-II, Filemaker was employed because it provides a simple and user-friendly user interface. In addition, there was already an archaeological data model (iDAI.field) which was implemented in Filemaker and maintained by the DAI. However, since Filemaker does not have any options for high-performance storage of 2D and 3D data, two PostGIS databases were used for the 2D and 3D data. The linking of a 3D model and the associated attributes were carried out using the ID connection tool. The copy to PostgreSQL became necessary in order to be able to efficiently query the data in the viewer. At that time Filemaker was not suitable as a performant database for a web application.</p>

Table 4: Data Management Components

User management:

Component name	Function
LDAP User Authentication	<p>Service to manage Roles and Login rights.</p> <p>Note: User Management is a very complex problem especially in the archaeological domain where many data are sensitive, e.g. looting. This leads to a challenge for scholarly collaboration for researchers who want to share data (on a somewhat large-scale) but cannot openly publish it. How can this challenge be addressed? In general two models are possible: (1) Centralized management where one authority checks if a user can be trusted and then provides access rights. (2) A decentralized approach where each producer of a new resource (e.g. 3D model) decides who gets access. This is an ongoing complex topic that requires institutional involvement not only for access but, for example, for copyright and other issues.</p> <p>Many existing repository solutions like Fedora usually allow the authentication via LDAP (Lightweight Directory Access Protocol)</p>

Table 5: User Management Components

Summary

Most of the components of the MayaArch3D System are still usable but need major revisions for updating dependencies. For example, the 3DSingleObjectViewer and 3DSceneViewer depend on the [GLScene.js](#) library, which itself depends on three.js. The version of three.js used in the MayaArch3D project was r63 whereas the current available version of three.js is r117. Those dependency changes must be integrated to be able to support newer formats (e.g.

glTF) that permit performance improvements, and are increasingly becoming more common. [This topic will be discussed in more detail in the Discussion & Recommendations section.]

Data & Data Formats

This section provides a description of the data and data formats produced in the MayaArch3D Project and integrated into the system.

Title	Description
Copan2013_LIDAR	Type: PointCloud Format: ASC, LAS Extent: Copan Valley (25 sq. km) Airborne LiDAR Scan by external company WSI
copan_lidar_dem_extended	Type: Raster Format: GEOTIFF Resolutions: 0.25 m, 1 m Extent: Copan Valley Created by Bruno-Kessler-Foundation (FBK) based on Copan2013_LIDAR
copan_hillshade_2013_extended	Type: Raster Format: GEOTIFF Resolution: 0.25 m Created by GIScience Heidelberg based on copan_lidar_dem_extended
copan_lidar_meshes_extended	Type: 3D Model Format: OBJ Resolutions: 6 different LODs from 60K triangles to 1920K triangles Input for 3D Tiles (Terrain)
3D Models FBK <ul style="list-style-type: none"> ● CRIA ● AMNH ● Temple 18 ● 2317_2630 ● Stela 11 	Single object photogrammetric, laser scanning or integrated 3D models-visualized in the single object viewer Formats: OBJ, three.js-JSON Created by FBK and optimized for web by Giscience Heidelberg

3D Models DAI	Hypothetical segmented reconstruction Temple 18 (10L-18) Format: three.js-SceneFormat, made from OBJ single segment models
3D Models FBK	Reality-based segmented Temple 18 Model (10L-18)
Copan 2014 DGPS	Positional stela data of the DGPS survey 2014: Created by: GIScience, FBK, DAI, IHAH Type: vector, CAD Format: shp, JPG, three.js-JSON Copyright and accessibility: pending Publication: pending Valley Stelas positions, 2D shapes and georeferenced 3D Models derived from DGPS measurements and CAD-based reconstruction by GIScience Heidelberg
structures_LODo FBK-UNL	3D Model of Copan structures based on 2D GIS footprints (PAC 1). Format: shp with type MultipolygonZ
Copy of Filemaker (iDAIField) Attribute DB	Attributes associated to LODo and segmented single models by ids Format: postgres SQL dump
2D Vector DB Backup	Format: postgres SQL dump with PostGIS specific data types
Segmented 3DGeometries DB Backup	Format: postgres SQL dump with PostGIS specific data types
DTM + Low LODs 3D DB	Terrain Tiles and LoDo No Backup dump of this data exists, it is only available inside the Windows Virtual Machine backup.

Table 6: Data Derived in the MayaArch3D System

The MayaArch3D system includes five broad data types: terrain, structures, attributes, dynamic texture, and stelae.

Terrain:

The terrain was acquired through airborne laser scanning (ALS), which resulted in point cloud data (**LAS** format). The data were post-processed into the widely used and stable raster format **GeoTIFF**--a format that the majority of geospatial software uses to ingest into the system as well as for long-term preservation and access. Additionally, the terrain was converted into several different Level-of-Detail (LOD) triangulated mesh formats. The data are available in the **OBJ** format--a well-adopted open format commonly used by archaeologists (and others). However, for purposes of browser-based 3D web visualization, the terrain was converted into the [three.js-JSON Model Format v3](#). While this was an unstable format, at the time (2014-2015) it was a cutting edge format that can be used within the visualization library GIScene.js / three.js. This three.js-JSON format is stored in the DTM+Low LODs database as terrain tiles. This format has been further developed since the end of the project and would need to be updated if the usage of three.js is an appropriate option for the future.

Nevertheless, the three stable formats (LAS,GeoTIFF and OBJ) still work in the majority of geospatial or 3D software, and newer formats could be derived from these original files if required for newer systems.

Structures:

The structures exist as two main types: auto generated LOD-0 models and manually derived hierarchically segmented models.

Autogenerated LOD-0 models

3D Buildings and freestanding monuments were generated by extruding 2D structure footprints (FBK, Giorgio Agugiaro) provided as a shapefile (**SHP**) (FBK-UNL, Heather Richards-Risetto). Structure base heights were set according to the terrain data (masl). The result is available as a 3D Shapefile (**SHP**). The geometry contains triangles with XYZ coordinates that are stored as MultipolygonZ data type.

For web-based visualization via a browser, the 3D shapefile was converted into the **three.js-JSON Model Format v3** and stored in the DTM+Low LODs database, which is not available as a backup SQL-dump. To reimport these dumps you need a PostgreSQL DB with the PostGIS Extension.

Manually-derived hierarchical segmented models

Manually-derived hierarchical segmented models were created for Temple 18 at Copan, Honduras and published using the 3DSingleObjectViewer.

One reality based model (laser scanned) was created and segmented (FBK, Fernandez and Suma) and are available as single segment models as **OBJ+MTL** (3d meshes with textures).

One hypothetical reconstruction was created by Mike Lyons and these segments are also available as **OBJ+MTL**.

These single segments of both models were converted into hierarchical scene graphs to provide parent child relationships for the segmented models and were stored as **three.js JSON Scene Format 3.2** to use online with a 3D engine (Figure 2).

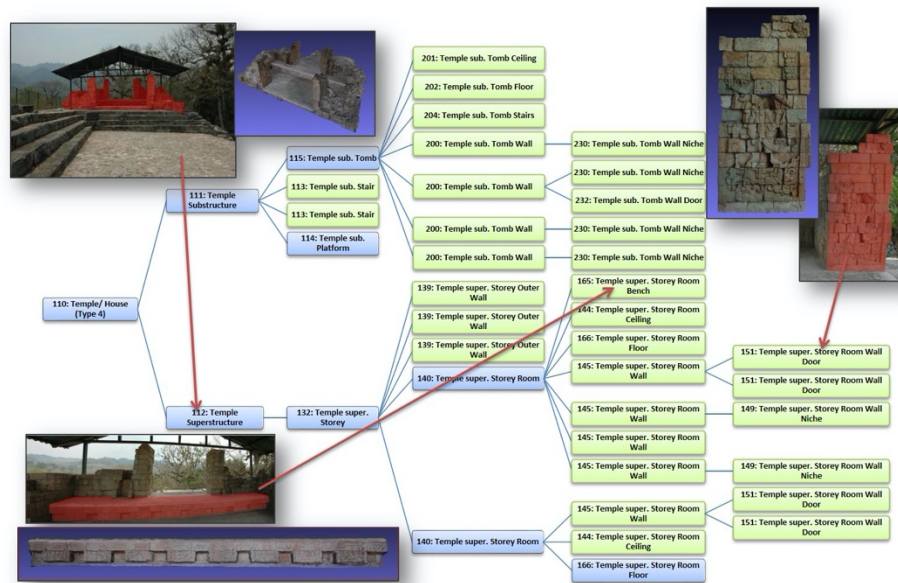


Figure 2: Hierarchical Scene Graph for Reality-based Model of Temple 18, Copan, Honduras (Courtesy: Mike Lyons, MayaArch3D Project)

Attributes:

All attributes relating to the models or model segments were stored in a separate database and referenced by id's in the models (JSON, SHP) or in the filenames of the models (OBJ). The attributes were managed by archaeologists with access to the i.DAIField (Filemaker) DB of the German Archaeological Institute. A copy of this DB was created and stored in a PostgreSQL DB, a **PostgreSQL SQL Backup file** exists and is currently stored at on a server at CDRH, UNL.

Dynamic texture overlays for the terrain:

In the 3DSceneViewer two overlay textures could be selected to be draped over the terrain in a dynamic manner. One was a grey-scale raster showing the intensity values of the return strength of the laser from the airborne LiDAR acquisition. The other was a colored hillshade. Both layers are stored as a **GeoTIFF** and were served in the MayaArch3D system by the Geoserver component using the OGC standard protocol OGC-WMS for dynamic tiling and format conversion to **PNG or JPEG**, such that they could be used as textures on the web.

DGPS derived Valley Stela Positions, CAD Plans, and 3D Models:

During the field 2014 campaign, we visited Copan's Valley Stelae (those outside the Main Group) to acquire data on their positions using a differential GPS (DGPS) and a laser distance measure tool to sketch CAD Plans. The DGPS data was processed using GNSS Solutions Software and resulting coordinates were exported as **CSV** and converted to a point shapefile (**SHP**). These DGPS ground points were used to geo-reference the CAD reconstructions from local configurations (Figure 3). Finally, we created georeferenced **SHP**, **DXF** drawings, **OBJ** and **three.js JSON Model** Formats of the Survey Points and CAD reconstructions. A draft report of the Survey is available, but unpublished so far, due to open questions about security issues regarding the exact positions of these monuments and possible risks of looting.

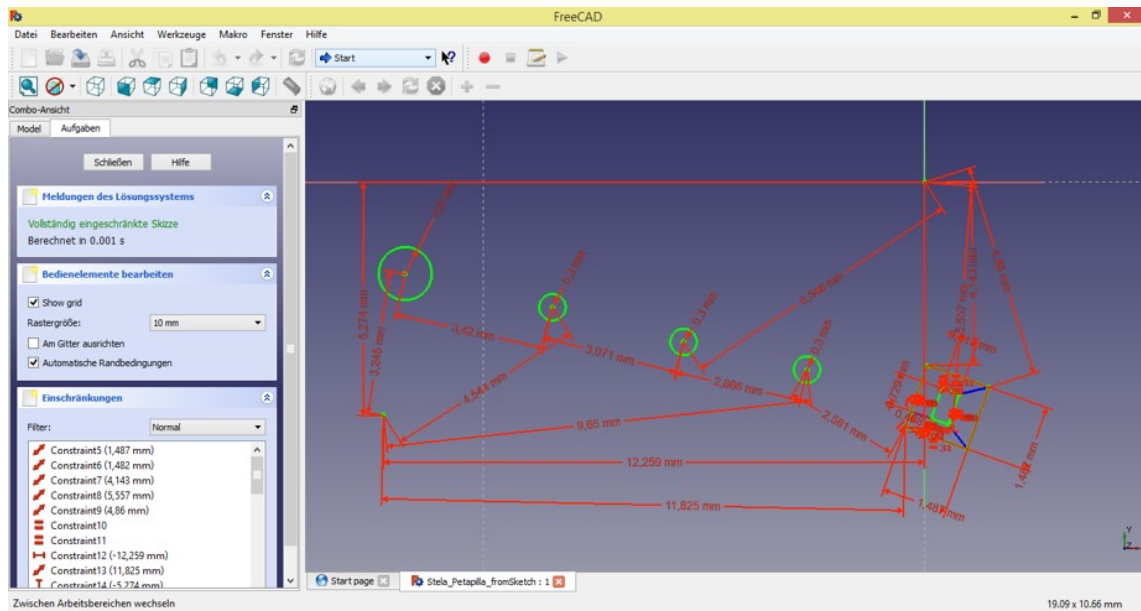


Figure 3: DGPS Points for Stela Petapilla, Copan, Honduras
(Courtesy: MayaArch3D Project)

Summary

All project data is still available. Most of the data is available in common, well-supported formats such as Shapefiles (SHP), GeoTIFF and OBJ. Application specific formats such as the three.js JSON Model format and three.js JSON Scene format, which were created for performance reasons and software dependencies, do not currently function on the web. However, they can still be updated to reinstate functionality. The content of the three databases (2d, 3d, and attributes) are available as zipped PostgreSQL SQL backup dump files.

The hardest data to recover data are the 3D Level-of-Detail Tiles that were stored in a separate database used by the W3DS Service. The data is still available inside the backup image of the Windows Server-based Virtual Machine (baru). To recover it, the VM must be re-deployed or alternatively the data could be regenerated from the available base data from which it was originally produced.

PART III: Discussion & Recommendations

This section provides an overview of possible solutions and replacements of components that could be used to revitalize the data and the functionality of the 3D WebGIS (i.e., 3D SceneViewer) and the 3D SingleObjectViewer of the MayaArch3D-II Project. Figure 4 serves as a guide for three recommended pathways.

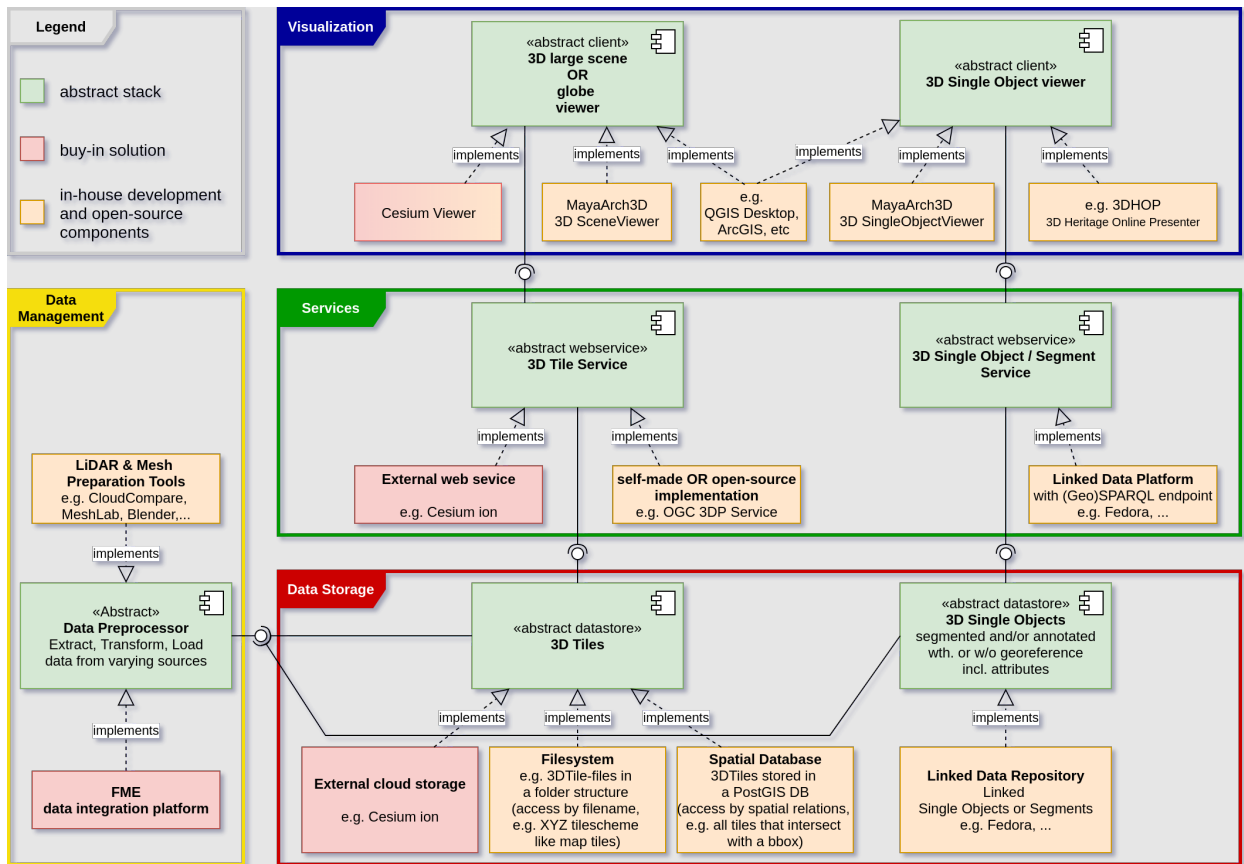


Figure 4: Recommended Pathways for 3D WebGIS Software Architecture

Figure 4 illustrates a 3-tier Architecture comprising Data Storage, Services (data access), and Visualization and Data Management. We have excluded User Management as it can be implemented in many ways, depend on selected components, and institutional policies and infrastructure. However, we discuss some recommendations related to this issue in the next two sections of the white paper.

Figure 4 also illustrates three implementation options for revitalizing the 3D WebGIS component (both 3D SceneViewer and 3D SingleObjectViewer) of the MayaArch3D system. The pathways include an abstract stack (illustrated in green) and two concrete implementations (illustrated in red and yellow). The abstract stack (green boxes) includes some parts of the infrastructure from the original MayaArch3D project. This is the case for large landscape scenes, which require tiled 3D data. This component requires a 3D tile service to stream varying scene resolutions to a client that understands and controls this kind of information structure (i.e., a scene or globe viewer). In contrast, the single object component (right) employs new developments for single or (individual) segmented 3D models. For example, Open-Linked-(Geo)-Data could be used to store and semantically enrich 3D Objects and their descriptions and related (meta-)data. This approach makes it possible to use

standardized ontologies such as [CIDOC-CRM](#) and others, which would increase data findability and retrieval, e.g., using federated queries, across repositories.

Across the 3 tiers several optional pathways of concrete implementations can be found. One path (left) gives options to implement a system for large landscape scenes, another path (right) shows options for single objects and potentially segmented models. Three types of concrete implementation options are proposed: (1) proprietary, non-free solutions (e.g. [Cesium Ion](#), [FME](#)), (2) free and open-source technologies combined with in-house development (e.g. updated MayaArch3D software), and (3) a mixed-approach using proprietary and open-source technologies.

Infrastructure Scenarios for landscape scenes

Dynamic web-based visualization and analysis of geo-referenced 3D landscapes (scenes), particularly in archaeology, is complex for various reasons such as large file sizes, data security, data findability, data access, lack of data standards and data interoperability, data heterogeneity, and complex spatial relationships among objects (von Schwerin et al. 2013). We present three infrastructure scenarios (options) for landscape scenes in 3D WebGIS. The first scenario is a buy-in solution that essentially replaces the MayaArch3D 3D SceneViewer, but allows the existing data to be brought “back to life” and provides a relatively expedient pathway to integrate additional data from researchers to more immediately facilitate comparative analysis across the Maya region. The second scenario revitalizes the 3D SceneViewer via in-house development and open-source solutions. While this option requires more upfront time and costs, its focus on adhering to recent 3D data standards and a lack of dependency on proprietary software offers longer-term stability and preservation as well as offering an infrastructure that can be customizable to many projects and cultural regions. The third scenario is a mixed-approach that combines in-house development with proprietary solutions to bring the 3D WebGIS component of the MayaArch3D project “back to life” sooner than scenario #2. It is less dependent on proprietary software and can be more easily modified to adjust to unexpected challenges (e.g. proprietary software no longer offered/supported or exorbitant price hikes) because only one part of the 3D WebGIS would be dependent on an external source.

(1) Buy-In solution

This scenario centers on the Cesium software. Cesium provides various workflows that support the integration of other 3D software (e.g., 3ds Max, Blender, FME, STK, WebODM). These workflows are well documented and can be found on the Cesium website.

The solution presented here contains two central components.

1. The Cesium Platform is an end-to-end web platform (software stack) that offers a variety of options for storing 3D data in the cloud, combining it with existing data, and then visualizing and analyzing it. This approach involves the least implementation effort. Only the integration of the client into self-hosted websites requires programming. [A tutorial can be found on the Cesium website](#)
2. The FME (Feature Manipulation Engine), a proprietary (non-free) data integration platform developed by Safe SoftwareTM. It provides various ETL (Extract Transform Load) processes for geospatial data.

Besides the visualization of 3D data Cesium provides the [Cesium ion SDK \(Software Development Kit\)](#). The Cesium ion SDK JavaScript library extends the open source CesiumJS JavaScript library and includes GPU-accelerated 3D analysis tools for measurement, sensor geometries, and visibility analysis as well as clipping planes and transform editors.

The advantage of using the proposed scenario is that all components and functionality for extracting, loading, transforming data, and data management (cesium cloud) already exist and can be directly used. A client website would still need to be developed in order to integrate the open source Cesium viewer; however, the [cesium.js](#) JavaScript library provides the necessary tools to integrate the 3D data into a website. [Cesium.js is designed so that it can be connected to other tools and platforms. Cesium.js uses WebGL in order to provide hardware acceleration for the visualization of 3D data and is licensed under Apache 2.0.] In contrast to the W3DS used in the MayaArch3D Project, Cesium offers the option of delivering tiles according to the OGC 3D Tiles standard (uses gLTF—a standard file format for three-dimensional scenes and models). Cesium.js, introduced by Cesium in 2015, is a community standard at the OGC. Additionally, Cesium.js supports spatial reference and could also replace the GIScene library (from MayaArch3D). The advantage here is that, in contrast to GIScene, Cesium.js is further developed and maintained.

The disadvantages of using a total buy-in solution is that it is based on proprietary software with a storage limit of 250GB for monthly subscriptions (\$499/month) or unknown pricing for custom solutions, black box processing for some components, limited extensibility, limited data findability, limited interoperability with other “repositories” for comparative analysis, and the company could cease to operate. We do not suggest this as a long-term solution.

(2) Open Source: In-house development

The second scenario is an open source solution and revitalizes the 3D SceneViewer via in-house development and newer open-source solutions.

The 3D tiles can be saved either directly in the file system or in a spatial database. On the server side, this data must then be provided by a service to be requested by a client. In contrast to the buy-in solution already presented, this server-side component must be implemented. The existing W3DS implementation no longer makes sense for this application because the OGC adopted the newer [3DP service as the standard](#). The [OGC 3DP](#) is a web APIC specification that is meant to be a standardized way to retrieve 3D data for visualization, like OGC WMS is for 2D maps. At this time, we are not aware of any free to use open-source implementations that support this standard. It combines two approaches tested the years before W3DS (for thick clients) and OGC Web Perspective View Service (WPVS) for thin clients. A [reference implementation](#) from the HFT Stuttgart does exist but it is unlikely that it is possible to use this code in a production environment without further adaptations.

In the MayaArch3D project, we developed the GIScene software based on three.js. As already shown above, three.js does not natively offer support for spatial references. This support on top of three.js was developed in the course of the project in form of the GIScene library. As already mentioned the use of the W3DS is not recommended because the OSG 3DP now exists. Given that the GIScene library requests the 3D tiles from the W3DS, to update the MayaArch3D system, it would necessary to adapt the GIScene code and GIScene library to the 3DP standard. Additionally, we expect that other system components would also have to be modified; thus, compared to a solution, which includes the use of 3DTiles and Cesium.js. Overall, while a fully open-source approach is ideal and would be a long-term goal, it likely to be much more complex and time-consuming, and thus we currently advocate a mixed approach.

(3) Mixed-approach: in-house development with proprietary solutions

The third scenario integrates in-house development with proprietary solutions. This approach offers a way to take advantage of standards, requires less intensive development than fully open source options and yet takes advantage of standards that facilitate data interoperability and sustainability. The approach still requires ongoing maintenance but to a lesser degree than the fully open source solution. Importantly, it provides a faster pathway to get the MayaArch3D system online again without resorting to a fully proprietary solution, and as a consequence continues to support the development of open source 3D WebGIS for humanities research.

In this scenario, the [Cesium Viewer](#) (open source) would replace the 3DsceneViewer. Some development is necessary to implement the Cesium Viewer, particularly for additional analytical functionality; however, given that the viewer uses the open standard Cesium.js, it does not exclude the use of customized or other open-source components as long as they conform to the standard.

Additionally, we suggest (at least in the short-term) employing Cesium Ion, a cloud service to preprocess and optimize 3D georeferenced data, as the terrain tiling component of the MayaArch3D Project is unstable and requires a major overhaul. While Cesium Ion does require a subscription fee (based on storage), it [tiles the terrain to the 3DTiles standard](#) and the output data is interoperable across visualization platforms. Additionally, it supports heterogeneous data such as meshes and point clouds and open standards including COLLADA (.dae), Wavefront OBJ (.obj), glTF (.gltf, .glb), and LAS (.las, .laz) files. A major disadvantage is the monthly subscription cost for the tiling service, and while the website states that the Cesium Ion platform can serve 3D Tiles to any compatible client, it isn't clear whether the 3D tiles can be directly downloaded for use in another geospatial platform. This is important because although the Cesium Viewer is free and open source, it appears that the monthly paid subscription is required for storing and serving the terrain to the viewer.

In terms of performing web-based analysis, the MayaArch3D SceneViewer provided a browser-based 3D Line-of-Sight Analysis tool based on WebGL and HTML5 (Auer and Zipf 2018) (Figure 5). The Cesium Viewer also offers browser-based analytical tools for visibility analysis. Additional analytical tools are available for the [Cesium viewer](#) but most require additional development for implementation.

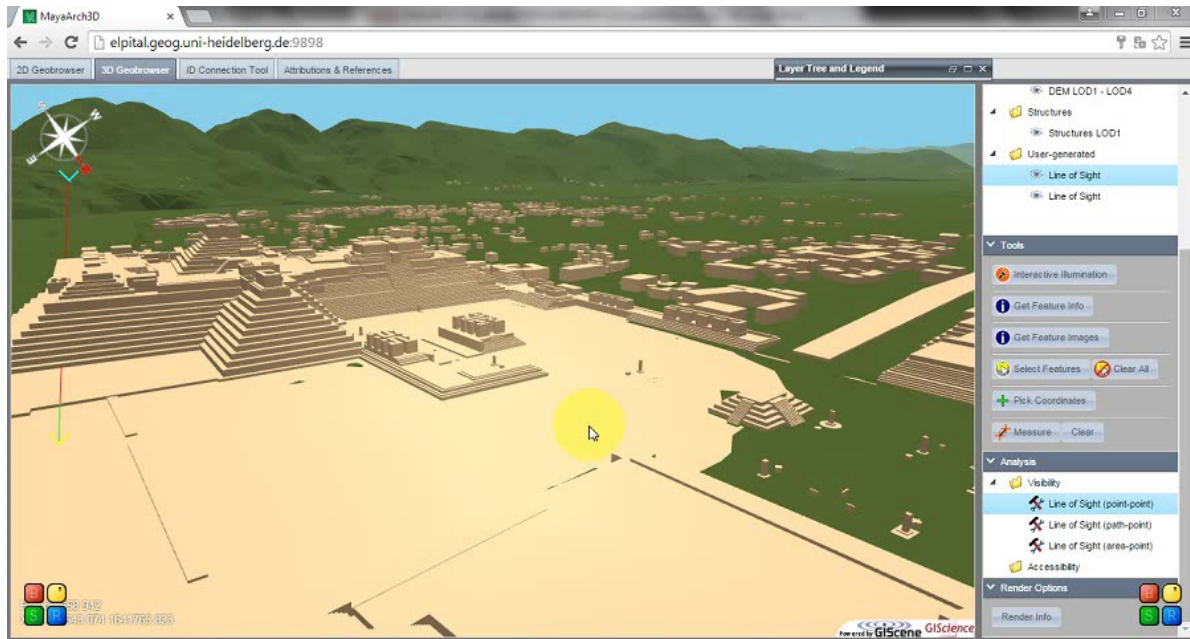


Figure 5: MayaArch3D SceneViewer illustrating Line of Sight tool

GIS desktop applications

While the above three scenarios can be developed to perform web-based analysis, it does not always make sense to carry out every type of analysis on the web or implement certain functionalities in a web environment—in some cases a local solution for analysis should be considered. To start, access to high-speed broadband is limited in many parts of the world (particularly the Global South). Additionally, the 3D data need to be optimized for visualization on the web, but decimating/optimizing 3D data for analysis is often problematic, particularly for geometric and metric operations. Therefore, it is important to also design and maintain a pipeline that connects GIS desktop application to the 3D WebGIS (regardless of the three proposed scenarios). Since version 3.0 the QGIS developer community has been working on the integration of 3D visualization and analysis functionality in QGIS—a free and open source GIS—([QGIS 3D Capabilities](#)). It is expected that support of the 3D Tile format in QGIS will be provided in the near future. However, we do not suggest that the use of a desktop (local solution) for some types of 3D geospatial analysis does not forgo the need for an open source 3D WebGIS for visualization or many types of analysis and in fact, we contend that it serves as an important mechanism to promote interdisciplinary collaborative research and public engagement with the spatial humanities.

Scenarios for single objects

While the focus of the Tier I NEH Digital Humanities Advancement Grant on “Revitalizing and Enhancing the Open Source 3D WebGIS of MayaArch3D” was the 3D SceneViewer, we also evaluated the 3D SingleObjectViewer. We decided to also carry out this research because we realized (even prior to any research) that to restore it online would require much less time and effort than the 3D SceneViewer. Moreover, given that other open source 3D Single Object Viewers such as 3DHOP, [Potree](#), etc. exist (in contrast to 3D Scene Viewers), our goal for the 3D SingleObjectViewer differed. We simply sought to get the tool to function online again to provide access to the data, rather than to research alternative and updated strategies to adhere to new standards, facilitate interoperability, etc. as was the goal for the 3D SceneViewer.

Currently, the 3D SingleObjectViewer is functioning online with the exception of actively linking to the database for segmented queries. It was recovered alongside the MayaArch3D WordPress website, which was moved to be hosted on a server at the CDRH. To get the tool running required reviewing and changing some URLs pointing to the viewer, and models were updated corresponding to their new paths on the CDRH server. Before publishing the recovered tool, we reviewed the versions of its JavaScript libraries to ensure the tool was as secure as possible with minimal effort. The jQuery and jQuery UI libraries were very easy to update with only minor changes to the page’s CSS required. Updating the three.js 3D rendering library was more complicated. The project provides a migration guide which describes changes necessary to one’s code for each release. The 3D model files used to demonstrate the tool’s capability are stored in an older JSON format which the three.js library no longer supports, so we began by trying the last version which supported the models. We worked our way backwards from there trying to make a few simpler search and replace fixes to test each version and arrived at what we determined to be a reasonable version to publish with. Further recovery would require more time and familiarity with the three.js library. More detailed documentation of the process is included with the code on GitHub part of a new [MayaArch3D GitHub organization](#) created for housing MayaArch3D development going forward.

We do, however, have some suggestions to improve the 3D SingleObjectViewer to make the data more findable and accessible. [Our suggestions build on findings from a NEH Preservation and Access, Research & Development Tier I Grant for “[Keeping Data Alive: Supporting reuse & repurposing of 3D data in the humanities](#)” (PR-253389-17).]

The original (and current) 3D SingleObjectViewer uses PostgreSQL/PostGIS—a relational database management system (RDBMS)—as storage solution. A disadvantage of this

approach is a restrictive attribute schema where data was stored as entries in a structure-representation-geometry relation with the same set of predefined attributes and then linked to the external IDAI-Field Filemaker AttributeDB (see Figure 6). To overcome limitations, we suggest to store the single objects as linked data with corresponding metadata and attributes in a self-hosted Linked Data Platform that uses a Resource Description Framework (RDF) triplestore—a graph database that supports optional schema models, i.e., ontologies. By combining linked data with RDF triplestore, users would have the following advantages to the current system: (1) query data outside the PostgreSQL database, (2) query data, (3) obtain highly relevant search results, and (4) query diverse data from different sources.

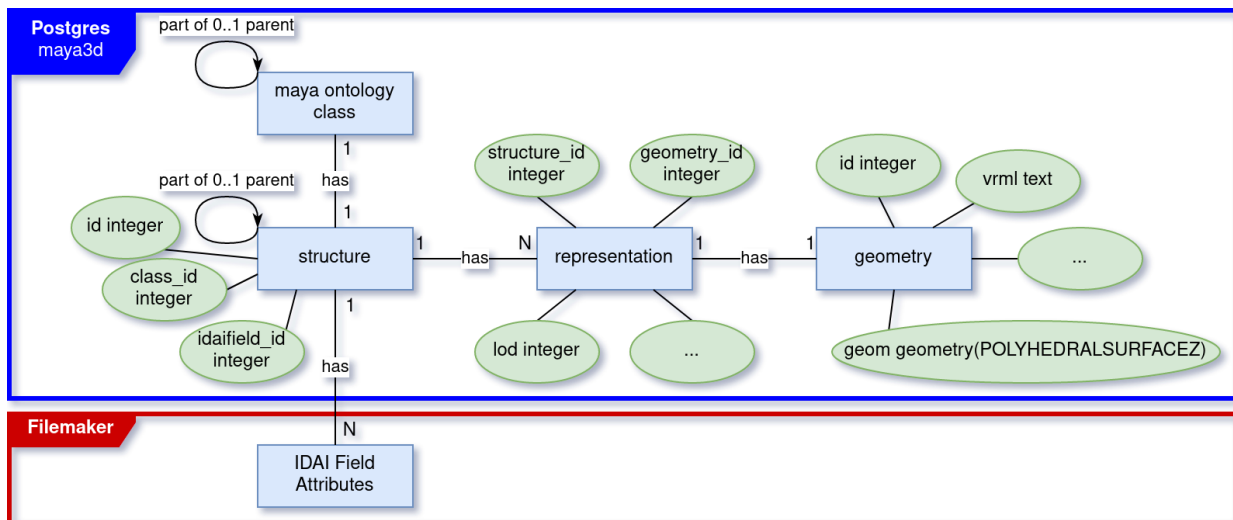


Figure 6: Schema of original 3D SingleObjectViewer of MayaArch3D System

[Nishanbaev et al. 2019](#) describe several Linked Data Platforms and triplestore databases that can be employed to support georeferenced data and 3D models. A spatio-temporal extension of CIDOC-CRM, the [CRMgeo](#), can be used to combine Geodata Standards from the OGC with the Ontology Standard of Cultural Heritage Collections to describe their spatial properties. Such Linked Data Platforms or Geospatial Semantic Web Platforms support several options for data access. These include: (1) provide a web-based user interface, (2) provide web-endpoints that are queried with a special query language called SPARQL or with spatio-temporal extension GeoSPARQL or stSPARQL and, (3) provide web-APIs to get the data into other applications, e.g. a 3D Model Viewer.

After a platform is selected that organizes the storage of single or segmented models and provides web access to them, a frontend is needed to visualize the models as well as allow

users to interact with them. We suggest two general frontend options beyond using the MayaArch3D SingleObjectViewer.

The first option is the simplest and quickest to implement, however, it only supports single objects that are not segmented. Unsegmented single object models created in the MayaArch3D Project can be visualized using existing open source 3D presenters such as [3DHOP](#) where the code base is actively maintained by the community. This approach is also an advantage in terms of low development and maintenance costs.

The second option focuses on a frontend that supports both simple single objects as well as hierarchical segmented models. Because of the complex data model of semantic hierarchically segmented models, there is no standard viewer available that could replace one-to-one the functionality of the **MayaArch3D-SingleObjectViewer**. This Viewer is based on the open-source library [GIScene.js](#), which is based on the popular and actively developed 3D library three.js. However, while three.js has continued to develop and be supported since the end of the MayaArch3D project, GIScene.js has not. Thus, future directions in this area necessitate considering whether to: (1) invest in the existing MayaArch3D software to develop updated and improved versions by updating the GIScene library or by implementing a new 3D library or (2) invest in developing an extension for another open source 3D Viewer such as 3DHOP that can also handle hierarchical segmented models.

Another potential alternative to geometric segmentation of models is to employ annotation (a function already supported in 3DHOP) to link additional information to specific subparts of 3D models. [Annotation is discussed in further detail below.]

Future Direction

The main goal of the Level 1 project on “Revitalizing and Enhancing the Open Source 3D WebGIS of the MayaArch3D Project” was to evaluate the 3D WebGIS software in order to develop a plan to bring the software back online in a more extensible, better documented, more accessible, and easier to maintain form. While much of our research focused on technological issues, we realize that software solutions for answering scientific questions cannot be evaluated in isolation. Therefore, in addition to reviewing the software code, our evaluation and recommendations also took into account approaches and concepts for the acquisition, management, analysis and visualization of archaeological data developed in interdisciplinary cooperation. In particular, we did not focus simply on how to get the MayaArch3D 3D WebGIS back online but rather on how to bring it back to life in a sustainable way with more efficient opportunities for customization to other projects. Six years have

passed since the end of the MayaArch3D project (ended in 2015) and many of the technologies used in the 3D WebGIS have been further developed and new 3D geospatial data standards have emerged. Changing and emerging technology will continue to be a challenge for the digital humanities (and beyond), and such changes are particularly relevant with web-based technologies, which are under greater development pressure than desktop solutions. In order to develop and maintain an open source 3D WebGIS require not only software modification but situating development and implementation within a broader perspective of data preservation, access, and reuse.

Implementation Overviews

Now that we have a good understanding of how we might begin to reconstruct MayaArch3D from a technical standpoint, we need to think about the specifics of implementation, both technical and socio-economic. To do that, we will look at the following areas: Data Packaging, Data Findability, Data Viewing and Analysis, and Data Creation, Manipulation, and Annotation. Note that the grants that funded the original MayaArch3D were mostly concerned with the last item, data manipulation and analysis, and any database structures created were in service to that aim. We now recognize that without a strong data access and storage plan in place, the manipulation step is not as useful because the underlying data can't be tracked or verified, and it is more difficult to add data. Therefore a focus on storage, packaging and findability must happen alongside manipulation and analysis.

Data access and findability is the most time sensitive concern at the moment, but as we will show below it is wrapped up in issues of data preservation and packaging. Without a central place to look for models, data, and metadata, it's likely that scholars will repeat work, sometimes at great expense. Scholars will also lose out on potential partnerships, and scholars at less well funded institutions will not be able to use data provided by wealthier institutions. In our [Keeping Data Alive Whitepaper](#), we recommend the use of an institutional repository for both preserving and displaying this data and recommend Fedora in particular due to its RDF capabilities. A bit more research is needed, but three ways an institutional repository might be implemented are:

1. One institution or organization committing to hosting a repository in perpetuity.
2. Multiple institutions commit to hosting a repository with cross references for findability.
3. Multiple institutions hosting repositories with one organization or institution hosting a central index to facilitate findability.

Sensitive data is also a concern: with archeological sites vulnerable to looting, it is important that some data be described but not made directly available, or described in such a way that would reveal sensitive information. A final concern in this area is potential size of the repositories, if users are able to create multiple versions of large data models.

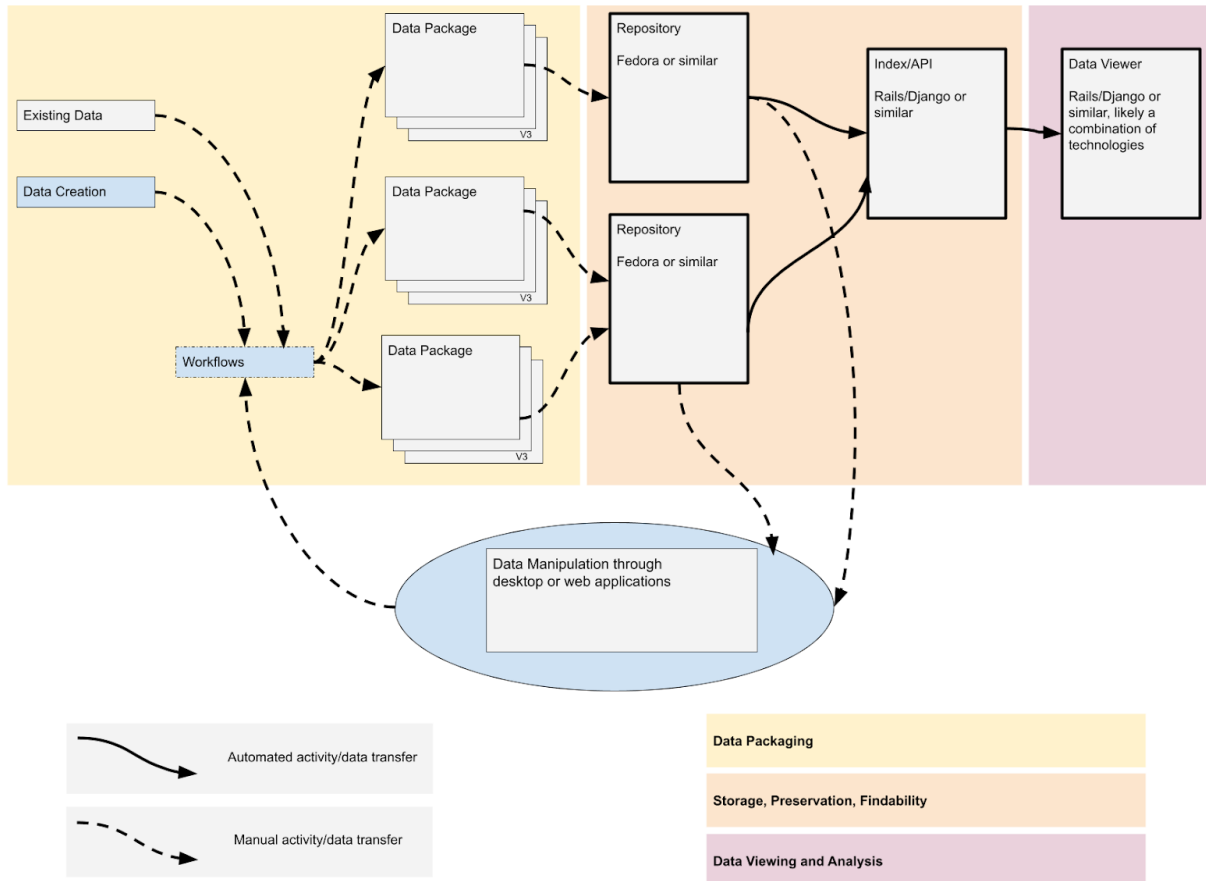
Closely linked to the issue of data access is data preservation. As mentioned above, preservation could be handled through the use of a repository that both provides access and long term preservation. If this is not possible, it may be necessary to separate these two concerns by designing one system for access and another for preservation (the original MayaArch3D was designed with access, not preservation, in mind). However, if this is the case, it must be explicitly stated what the preservation component of the equation is. Which models and metadata must be preserved? Is paradata part of that preservation?

A final aspect having to do with data access and preservation is that of formats and included metadata. Again, we have done some of this pre-thinking in the Keeping Data Alive whitepaper, choosing the Europeana Data Model (EDM) for metadata for models. More thinking is needed on this front because there are still more complex metadata and paradata issues that may not be handled by the EDM. In this case, [as the technical recommendations describe](#), it may be necessary to provide more elements from the CIDOC CRM metadata scheme. Some of the CIDOC elements are in the EDM, but not all of them, and not the ones most geared towards some of the finer points of Archeological models. In any case, the metadata fields needed for deposit into a repository need to be clearly defined and held in a central location, along with scripts/schemas to validate/check the data and a definition of file types.

Finally, once the data storage, preservation, formats, and metadata have been documented and workflows created to enable the deposit and retrieval of data, a manipulation layer can be reconstructed again. This would be based on the technical recommendations from Part 2 though further compartmentalizing may be needed to prioritize aspects of the work. Research from the Keeping Data Alive project indicate it may be possible to pull data directly from institutional repositories, however, that has only been explored as a proof of concept. Research needs to be done to make sure systems can support both the throughput needed and support for all components is in place. It seems likely that another, intermediary database or storage system would need to be built to ingest materials for dynamic manipulation. If this is the case and data sensitivity issues will need to be considered on the manipulation layer as well. In addition, given sensitive data concerns users will likely need to upload their own data to use alongside published sources. Policies must be in place regarding data use and retention in this case.

Recommendations

Given the above we are making the following recommendations for component implementations, likely split among 2-3 sub-projects.



Data Packaging

In order to share data, the first step is to package up the data with appropriate metadata, paradata, and textual description. When packaging data, users need to consider peer researchers and potential future users of the data, as well as the requirements of data repositories. Data packages will contain the data itself (likely in several formats), metadata, paradata, and prose documentation as to the collection and decisions made about the collection, suppositions, provenance and migration of data. Data packages would need to be able to relate to each other, with top level metadata in the Europeana Data Model, CIDOC CRM, or other well supported metadata scheme, along with crosswalks to produce any

additional metadata formats needed for deposit into identified data repository or repositories.

1: Determine ideal packages for various types of data (born digital, photogrammetry, etc)

This step in the process includes both listing the types of data and metadata, and including crosswalks between metadata as needed. Research is needed to create packages that conform to various existing repositories deposit rules. Crosswalking scripts could be modeled on concepts from [Frictionless Data](#) or other existing concepts. The goal is to produce all metadata up front so deposits into repositories will be as easy as possible.

Another essential aspect is ensuring the data that scholars have access to is in a format they can use. Where possible, this should mean an open source, documented format, but occasionally this may mean a well-documented and supported proprietary format, given the state of 3D and GIS applications.

2: Develop workflows to create data packages

In order to increase the use of our collections and metadata, there is a need to develop a packaging and dissemination model (comprising specific yet transferable workflows) that is as low barrier and frictionless as possible, while also advertising collections to the communities that could benefit from and reuse them. A data package format without detailed instructions on how to get to that format will see little adoption. Creating these workflows will involve gathering subject experts, librarians, end users, and researchers to create workflows that describe the creation of such data sets, including what expertise is required to complete each step. The work done by the [Collections as Data](#) group will be useful for this step.

3: Publish recommendations on data formats and workflows

Once research has been done on formats and workflows, the results must be published. This will likely come in two formats: a technical specification for the data itself, which defines what must be included, what is optional, and how to validate; and workflows. [Examples of data specifications](#) can be found on the frictionless data site, though these specifications are much simpler than 3D model specifications will be because they are mostly concerned with tabular data. Where to publish these specifications is an outstanding question as well. Workflows would be narrative in nature, with diagrams and tables to help with clarity.

Possibilities might include: joining the frictionless data group to add to their specifications, or creating a similar humanities focused site. For our own research this would have an archeological disciplinary focus, but a broader project might be creating a clearinghouse site along the lines of the [Linked Open Vocabularies site](#).

Data Storage and Preservation

Once data packages are described and documented, the next step is depositing them to ensure longevity. Many data repositories already exist; the majority are institutional or consortial, and others are disciplinary. In order to complete the next step, it is likely that disciplinary repositories focused on archeology will need to be identified that have the capacity to accept deposits, such as [The Digital Archaeological Record](#) (tDAR). Failing that, though, any repository that can accept a zip of the data with a minimal description that the researcher has access to should be identified for this step, which focuses on preserving the data for the long term while other options are explored. In the best case scenario, data will be deposited in multiple repositories (such as a disciplinary and an institutional repository) for added data preservation. As always, scholars affiliated with an institution should work with local librarians to consult on the best way to preserve data. More research is needed on ways for unaffiliated scholars to preserve their data.

In order to facilitate reuse, the next section will discuss finding preserved data.

Data Findability

We recommend the many repositories, one index approach, unless an organization or society can be found that can make a good permanent home for a repository that can expose the metadata needed. This approach would not specify the type of repository except requiring that it have API access to facilitate ingest of materials and a base set of metadata. The index would need to be maintained by an organization or university, but the overhead to maintaining it would be lower than hosting and maintaining an entire repository. A list of repositories to import from would be maintained, and every time a new repository is added some work is needed to set up index scripts. Cross repository indexes like [BASE](#) already exist for finding data, but the metadata indexed is too general to help a specialist narrow down useful content. In addition, BASE's metadata is sometimes spotty (Tay, 2018). A disciplinary index can draw from the metadata agreed on in the defined data package, and can index not only from the stock metadata the repository collects but discipline specific metadata in a text file. Such an index would allow for complex queries to narrow down needed data.

As a proof of concept there should be two to three repositories with working ingest scripts. The index would contain enough metadata to enable discipline specific searching, and will pull in data subject to repository API guidelines. For sensitive data, entries will be made for the data that scrub identifying information with a note adding about what has occurred. In this initial conceptualization it is unlikely that one organization can fully vet the data and who should have access, so it will likely remain up to the potential collaborators to correspond with the holder of the data, who will then need to determine the requirements to ethically share data. In this scheme, the repositories will provide both public access to data and long term preservation, and it will be up to the institutions maintaining the repositories to commit to this.

The index should be constructed in such a way that the data is exposed for computational uses through an API. This has several benefits, including the facilitation of data viewing and analysis applications, but also would allow researchers to download bulk data to compare metadata in ways the application would not allow.

Data Viewing and Analysis

During the Keeping Data Alive project (NEH-funded PR-253389-17) we envisioned a front end interface that pulled directly from a repository infrastructure to display

From the Keeping Data Alive whitepaper, under “[Activities and Accomplishments](#)”:

“One of the goals of the Keeping Data Alive grant [was] to design the infrastructure for a repository to ingest, host, deliver, and retrieve 3D models linked to metadata, paradata, and descriptive data that can be visualized and edited in an open source 3D visualization environment with changes tracked by the repository. ... Given that libraries are the main data stewards in the U.S. ... matching infrastructure commonly used in the libraries community was prioritized. We opted for a repository approach that can be customized, better meet libraries' requirements and potentially promote preservation and access efforts beyond individual projects. Along similar lines, building on existing capacity and infrastructure in the libraries community, the Research Description Framework (RDF) was selected because it offers a standard data structure for data interchange on the web, and is widely employed as a key component of linked data.”

Keeping Data Alive

- Home
- Browse 3D Objects
- Browse 3D Scenes
- Map
- About

3D Scene #1234 (placeholder name if other scene name is not supplied)

Text to describe model entry. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Fusce tincidunt leo et tempus sodales. Morbi id tempus lectus. Sed vulputate nunc eget nisi tincidunt ultricies. Mauris euismod ante nec odio dignissim euismod. Maecenas vehicula dolor urna, nec dignissim orci finibus eleifend.

Metadata about this Scene

Location: Mexico City, Mexico

Contributors to this scene

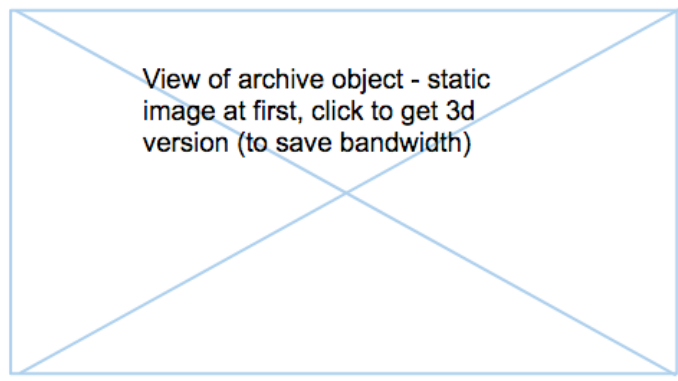
- [Heather Richards-Rissetto](#)
- [Jane Doe](#)

This scene contains these objects

- [Object 1](#)
- [Object 2](#)
- [Object 3](#)

Scenes

- May 18, 2017 - original by Heather Richards-Rissetto
- January 10, 2018 - uploaded by Jane Doe
- June 22, 2017 - uploaded by Jane Doe
- April 4, 2018 - uploaded by Jane Doe



[Switch to 3D Rendering](#)

Cite this derivative

Class aptent taciti sociosqu. ad litora torquent per conuia nostra, per inceptos himenaeos. Mauris euismod ante.
<https://keepingdataalive.com/scenes/scene1234/d1>

Data formats and downloads for this derivative

OBJ	STL	DAE	TXT
-----	-----	-----	-----

Wireframes from the [Keeping Data Alive Project](#). Caption: A representation of what the front end repository view for a scene might look like. On the left there is a title, which will need to be generated if one is not provided, a description, metadata about the scene such as location, contributors to the scene, and objects the scene contains. On the right there is a box for a view of the archive scene, which will be a static image replaced with a 3D representation on click to save bandwidth. Below that there is a clear citation for users of the resource, and an area for users to download data formats for the particular derivative objects: OBJ, STL, DAE, and Text files.

Our research during the MayaArch3D project complicated this view by exposing additional metadata we will need to consider, but these concepts should still provide a good basis for Data Viewing and Analysis.

Data Creation, Manipulation and Annotation

The final bit of functionality from MayaArch3D to replicate is the ability to create new data and manipulate objects and data. The implementation of this is discussed in detail above but we provide an overview below.

While all project data is still available in common, well-supported formats, they will need to be updated to reinstate functionality and as noted above should be packaged along with metadata in a reusable format. This will form the base of the system, whether linked to a repository or not.

Most of the components of the MayaArch3D System are still usable but need major revisions for updating dependencies. Those dependency changes must be integrated to be able to support newer formats that permit performance improvements, and are becoming more common.

There are three main Infrastructure Scenarios for 3D WebGIS landscape scenes: a Buy-In solution, an open source solution with In-house development, and a Mixed-approach: in-house development with proprietary solutions. Which of these is chosen depends on several factors: the relative cost effectiveness of a proprietary vs an open source solution, how likely it is that an open source solution will be reused in other contexts (which could affect the cost), and how much programmer time the host institution must devote to open source development. While our preference is for open source solutions, we recognize that it is likely we will have to go with a hybrid approach.

For single object models, we were able to get the [viewer back online](#) but without the underlying connections. The current 3D SingleObjectViewer uses a relational database management system (RDBMS) as its storage solution. A disadvantage of this approach is a

restrictive data model where data was stored as entries with predefined attributes. To overcome limitations, we suggest to store the single objects as linked data with corresponding metadata and attributes in a self-hosted Linked Data Platform that uses a RDF triple store—a graph database.

For both single object and 3D GIS viewing, it is essential that the project is able to make use of segmentation work already done while also allowing for annotation of unsegmented objects. There is current research into using machine learning to auto segment objects [Matrone et al. 2020]. In the meantime annotation has emerged as a commonly requested and implemented feature (see, for instance, the “3D model viewer features” table in [Champion, Erik, and Hafizur Rahaman, 2020]).

Ideally the entirety of the MayaArch3D redevelopment would be one large project, but if needed could be split into three sub-projects: the storage of existing data in a stable and queryable location, the querying of that data from the single object viewer, and finally the 3D GIS component which would be the most complex.

References

Auer, M. and Zipf, A.

2018. “3D WebGIS: From Visualization to Analysis. An Efficient Browser-Based 3D Line-of-Sight Analysis” *ISPRS Int. J. Geo-Inf.* 7, no. 7: 279. <https://doi.org/10.3390/ijgi7070279>

Billen, N., Loo L., Auer, M., Zipf, A. and H. Richards-Rissetto.

2013. “Development of a 4D WebGIS for Archaeological Research. Proceedings of 16th AGILE Conference on Geographic Information Science, Workshop on Integrating 4D, GIS and Cultural Heritage, Leuven, Belgium

Bodenhamer, D., Corrigan, J., & Harris, T. (Eds.).

2010. *The Spatial Humanities: GIS and the Future of Humanities Scholarship*. Indiana University Press.

Bodenhamer, David J., John Corrigan, and Trevor M. Harris.

2015. *Deep maps and spatial narratives*. Bloomington: Indiana University Press.

Champion, Erik, and Hafizur Rahaman

2020. Survey of 3D digital heritage repositories and platforms. *Virtual Archaeology Review* 11(23):1–15. DOI:[10.4995/var.2020.13226](https://doi.org/10.4995/var.2020.13226).

Gregory, Ian N., and A. Geddes.

2014. Toward spatial humanities: historical GIS and spatial history.

Kansa, S., Atici, L., Kansa, E., & Meadow, R.

2020. Archaeological Analysis in the Information Age: Guidelines for Maximizing the Reach, Comprehensiveness, and Longevity of Data. *Advances in Archaeological Practice*, 8(1), 40-52. doi:10.1017/aap.2019.

Kuroczynski, P.

2017. Virtual Research Environment for digital 3D reconstructions – Standards, thresholds and prospects. *Studies in Digital Heritage*, 1(2), 456–476.

<https://doi.org/10.14434/sdh.v1i2.23330>

Matrone, F., Grilli, E., Martini, M., Paolanti, M., Pierdicca, R., Remondino, F.

2020. Comparing machine and deep learning methods for large 3D heritage semantic segmentation. *ISPRS International Journal of Geo-Information* 9 (9): 535.

Doi:10.3390/ijgi9090535

Reindel, M., Zipf, A., von Schwerin, J., Auer, M., Loos, L., Fecher, F., Lyons, M., Stelson, L., Billen, N. 2016. Abschlussbericht des BMBF E-Humanities Verbundprojektes „MayaArch3D: Ein webbasiertes 3D-GIS zur Analyse der Archäologie von Copan, Honduras”

Richards-Rissetto, H. and J. von Schwerin.

2017. A Catch 22 of 3D Data Sustainability: Lessons in 3D Archaeological Data Management & Accessibility. *Journal of Digital Applications in Archaeology and Cultural Heritage*. 6: 38-48.

doi:10.1016/j.daach.2017.04.005.

Tay, Aaron

2018. Enhancing the discovery of institutional repository contents - 6 sources to consider.

<http://musingsaboutlibrarianship.blogspot.com/2018/03/enhancing-discovery-of-institutional.html>, accessed January 21, 2021.

von Schwerin, J., H. Richards-Rissetto, F. Remondino, M. Grazia Spera, M. Auer, N. Billen, L. Loos, and M. Reindel.

2016. Airborne LiDAR Acquisition, Post-Processing and Accuracy-Checking for a 3D WebGIS of Copan, Honduras. *Journal of Archaeological Science Reports* 5: 85-104. doi:10.1016/j.jasrep.2015.11.005.

von Schwerin, J., M. Lyons, L. Loos, N. Billen, M. Auer, and A. Zipf.

2017. Show Me the Data!: Structuring Archaeological Data to Deliver Interactive, Transparent 3D Reconstructions in a 3D WebGIS. In *3D Research Challenges in Cultural Heritage II : How to Manage Data and Knowledge Related to Interpretative Digital 3D Reconstructions of Cultural Heritage*, editors S. Munster, Pfarr-Harfst, Kuroczyński, Ioannides. Springer Int. Publishing.

von Schwerin J., Richards-Risetto H., Remondino F., Sauerbier M., Eisenbeiss H., Forte M., Gruen A.

2009. Digital Documentation and Reconstruction of an Ancient Maya Temple and Prototype of Internet GIS Database of Maya Architecture. Final Report to the NEH on the Digital Humanities Level I Start-Up Grant.

von Schwerin J., Richards-Risetto H., Remondino F., Agugiaro G., Forte M., Maqueda R.

2011. The MayaArch3D Project; Digital Technologies for Research in Maya Archaeology. Final Performance Report and White Paper for NEH Digital Humanities Level II Start- Up Grant.

Wright, H. & J. Richards

2018 Reflections on Collaborative Archaeology and Large-Scale Online Research, *Journal of Field Archaeology*, 43:sup1, S60-S67, DOI: [10.1080/00934690.2018.1511960](https://doi.org/10.1080/00934690.2018.1511960)