

2-2019

Using a Content Management System for Student Digital Humanities Projects: A Pilot Run

Amy E. Gay

Binghamton University--SUNY, aegay@binghamton.edu

Follow this and additional works at: https://orb.binghamton.edu/librarian_fac



Part of the [Digital Humanities Commons](#), and the [History Commons](#)

Recommended Citation

Gay, Amy E., "Using a Content Management System for Student Digital Humanities Projects: A Pilot Run" (2019). *Library Scholarship*. 46.

https://orb.binghamton.edu/librarian_fac/46

This Article is brought to you for free and open access by the University Libraries at The Open Repository @ Binghamton (The ORB). It has been accepted for inclusion in Library Scholarship by an authorized administrator of The Open Repository @ Binghamton (The ORB). For more information, please contact ORB@binghamton.edu.

Using a Content Management System for Student Digital Humanities Projects: A Pilot Run
Amy Gay - Digital Scholarship Librarian, Binghamton University

Content Management Systems

Content management systems (CMS), a phrase that is defined pretty much how it reads--they are systems that manage digital content. If you have worked within a library's archives, special collections, history and genealogy department, or within museums, you have probably used one either on the front end or managed digital collections on the backend. Commonly used content management systems include [Wordpress](#), [Drupal](#), [ContentDM](#), [ArchivesSpace](#), [Omeka](#), and many others. With the ever-growing number of open source and purchase options for managing digital content, it is important to consider the following:

- Types of items within the collection (images, documents, videos, etc.)
- Size of items (can the CMS handle the upload of large file sizes)
- Budget--is it worth purchasing a CMS or do the open source options meet the needs of the stakeholders and collections
- What features are needed to display collections (i.e. metadata schemas, plug-ins, etc)

For this use case, I will share a pilot project using a newly developed content management system, Omeka S, for an undergraduate History class' digital humanities projects.

Omeka Classic and Omeka S

Omeka is a free, open source content management system developed by the Roy Rosenzweig Center for History and New Media at George Mason University. Currently, there are two versions of Omeka: [Omeka Classic](#) and [Omeka S](#). Omeka Classic is used by institutions worldwide for all types of collections--images, large files, interactive maps and timelines, etc. OmekaS is a newly developed platform with future developments still coming.

At Binghamton University, we currently use Omeka Classic for our digitized materials, such as manuscripts, photographs, postcards, and oral history audio files. While we currently plan to continue using Omeka Classic for our digitized special collections, there was an interest in learning more about the capabilities and uses for Omeka's newest development, Omeka S.

There has been interest by faculty to use Omeka Classic for their own digital collections from research projects and for integrating more hands-on, visual, and publicly accessible student projects that students can then showcase the work they have done. With Omeka Classic is designed as a single instance around a general theme, it may be difficult to draw the connections between our digitized special collections and the variety of collections and projects created by faculty and students. Since Omeka S is meant to have multiple sites with a shared backend, our Systems and Digital Initiatives team felt we should test out its capabilities even at its beginning development stage. As Digital Scholarship Librarian, I took lead on learning how to use Omeka S and figuring out what it could be used for related to digital scholarship and projects for faculty and students.

Background

Adding Omeka S took the assistance of two other members of our Library Systems team to help with the download to our server, knowledge of [MySQL](#), and following the instructions on their [GitHub page](#). Overall, the process ran smooth and was installed within a couple hours.

After installation and account creation, I read through the [Omeka S User Manual](#) to get a feel of the backend and view the platform's features at its basic form. Overall, I found their user manual to be a good resource to have on hand while navigating through the backend of Omeka S. The navigation is fairly user-friendly, more so now that newer version updates have helped clean up bugs that caused issues at the item level. It is important to keep in mind that Omeka S is still in early development (its current version is 1.2.0) and their team is working to fix other bugs and are quick to handle issues that are shared on their GitHub page and forum. To learn more about what is currently happening with issues / updates, check out their [forum](#). Users of Omeka S also have the ability to create their own custom themes and modules to share with the larger Omeka S community. Themes and modules that have currently been added can be viewed here: [Themes & Modules](#).

Omeka S also has multiple vocabularies available for collections, including Bibliographic Ontology, Dublin Core, Dublin Core Type, and Friend of a Friend. As I am not as familiar with Bibliographic Ontology and Friend of a Friend, these are two I will need to learn more about their uses and when they would be more applicable than Dublin Core.

For people who currently manage the backend of Omeka Classic, you will notice some similarities; however, the modules, or plug-ins, are labeled different and have different sorts of capabilities. Currently, Omeka Classic has a plug-in called [Neatline](#), which allows users to create interactive maps and timelines--a plug-in that I hope eventually becomes a module for Omeka S as this has been specifically mentioned for interest by colleagues and faculty.

Pilot Project

I started in my role as Digital Scholarship Librarian a few days before the start of Spring semester. When I arrived, I was informed there was a History professor teaching a local history class that semester who was interested in piloting digital student projects using Omeka S. Due to time constraints of the project and lack of knowledge on Omeka S at that time, I decided to keep Omeka S at its basic level with no added modules for this pilot. In some of the feedback and issues submitted to the GitHub page and forum, I have read that using different modules with different instances can have implications with the other instances and although I would like to "break" our instances on my own time for learning purposes, it was best to play it safe so student projects had a greater success rate of being built and showcased for this pilot.

The professor and I met twice prior to bringing the students on-board. During our first meeting, we discussed the goals for the projects, the capabilities of Omeka S at its basic level, his role and responsibilities versus my role and my responsibilities, learning outcomes for students, what types of digital items will be part of the projects, a timeline for completion, and the audience

(particularly whether it would be a private site or viewable by the public). In our second meeting, I set him up with an account and showed him how to navigate around the backend. I gave him permissions set at the Author level, which allows him the following permissions:

- The author is able to search, read, and create items, item sets, media, and site pages. S/he can only edit or delete content which the user has created.
- The author has search and read only privileges for vocabularies and sites.
- No privileges for adding modules (which was not an applicable need for this site).

He spent some time learning the backend of the site and reached out to me via email when questions arose.

About two months before the project was due, the professor had me attend a class session where I taught students how to build their projects in Omeka S and demo examples of what different setups for how items on their pages and the metadata can look like. The class had about fourteen students, all of which were undergraduate upperclassmen. The professor and I had built a customized metadata template for their site using Dublin Core fields. I brought the students a handout with the fields and their definitions--another perk of Omeka S is that when you hover your mouse over each field, the definition will appear and there are additional details listed regarding file types that can be uploaded to those fields and text boxes for manual written descriptions. I found this feature to be helpful for the students since only a few of them were familiar with metadata and digital collections before taking this course. While going over each of the fields, students had questions regarding what "Rights Statements" were so we spent some time looking at rights statements from different institutions and had a discussion on Creative Commons. The professor had example items for them to fill out the fields for on paper and we walked around to help and answer questions.

The following day I added the students to the Omeka S site as users, with the same permission levels as the professor (Author). I checked in with the professor to see if the students received their confirmation emails but those who replied to him had said no. After giving it two days and still no confirmation emails, I met with our Library IT to see if there we could find the step I was missing as I did not have a problem when adding the professor earlier in the semester. We looked through my permissions as Global Administrator at the system level and Administrator at the site level and everything looked okay. We tried deleting my account and re-adding me and still did not work. After looking through what else had changed, we realized that it was because we changed the domain name for our Omeka S page and because I still had it bookmarked to go to the old address, it was not responding properly. After re-bookmarking the site under its new name, I re-added each student and they all received their confirmation emails--so far, we have not found a way to batch upload a class list for users and each name must be added individually.

Although we originally planned to hold off on updating Omeka S until this project had completed, I learned about one bug in particular that would hinder students work--this bug would make it so that any item added could not be deleted if it was a mistake. It would just sit on the page as an unfinished item and add clutter / potential frustration for students who have never done this work

before. Therefore, after making the professor aware of the potential concerns with doing so and getting his okay, a member of our library systems team updated the version of Omeka S on the server with no issues and the bug was immediately fixed. Students worked on their final projects with very little communication of questions / concern. To visit the students' final projects, you can do so here: <https://dhprojects.binghamton.edu/s/BingBiz/page/home>. Although at its basic level with no added modules, the student pages still came out well done and have the potential for enhancements by the modules and themes being added.

Lessons Learned

User Permissions

There are user permissions at the site level and at the systems level. At the systems level, I made the professor and students all Authors because anything higher would give them too many permission on other sites that are created besides their own, based on the [User Manual](#). At the site level, I made the professor an Administrator, which gave him the same permissions as I had at that level, and made students Editors which gave them permission to add and edit content. I hope that with future installments of Omeka S there are additional permission levels created at both the system level and site level, as there is currently a drastic difference between what can be done at each levels currently. For example, at the site level, I could make students a Viewer, Editor, or Administrator--where Viewer can only view the pages but not work on them but Editor is able to add and edit items on their pages and the pages of others within the site. I feel there needs to be a Page Editor type role between Viewer and Editor so that students do not accidentally change content on the page of another student.

Student knowledge level

One thing I wish I discussed with the faculty member prior to meeting with students is the knowledge level of the students regarding special collections, digital collections, metadata, etc. This class had very little knowledge of content management systems and metadata schemas so I ended up taking a few steps back and breaking things down more during class. In retrospect, I wish we had more than one session together to give them a better foundation before building their pages. This will be especially important once I start adding modules and different themes to be used.

Future uses

Over the next few semesters, I plan to add themes and modules to Omeka S and see what happens when multiple site instances are running at once with different modules being used. I would like for this tool to be available for future classes by Fall 2020.