Introduction

Hello, my name is Caroline and I am a GIS Specialist and cartographer for the Wisconsin Geological and Natural History survey. I will be talking about our project to move previously-published maps into the Geologic Map Schema, or GeMS.
These slides and other resources are available in this GitHub repository. I hope this will be useful for anyone doing this in the future, for instance if you received NGGDPP funding for Priority 2 this year.

[slide: bedrock map] Last year, I attended the GeMS workshop before DMT. I brought some datasets with me for some of our regional bedrock maps, to try to follow along;

[slide: notebook scan] however, the schema of our initial data just didn't fit with GeMS very well. I got stuck trying to detangle which lines belong in which layer, and how attributes would transfer.

[slide: point a/point B] In the year since, we have spent some time moving several previously-published maps into GeMS, so I would like to share with you an overview, some challenges, and decision points in our workflow.

Our starting point

[slide: map images] We worked with 21 maps, originally published between 1984 (NE Wisconsin Region) and 2011 (Brown County). Some were bedrock and some were surficial. The linework had already been digitized as vectors, but features were organized differently, and attributes were captured differently: in some cases using subtypes and coded domains. We worked with counties at 1 to 100,000 and some regions at 1 to 250,000 scale.

[slide: website image] Some of these maps had data available for download from our website in coverage format, downloaded as interchange files, some were available as a georeferenced image, while some maps offered only a non-geo PDF. Our priority was to improve the data products that we offer for download.

[slide: map of the maps/ data availability] we prioritized maps with data in coverage format or maps whose vector data is not available currently on our website.

We hired one student, Alex Cassell, to work with me, and she spent about 200 hours over the course of about 30 weeks exclusively on this project.

Developing a workflow

[slide: aspects] We worked through the requirements in this order, taking care of the geometry and attributes first and just barely reaching the metadata and symbology stages. So this talk will focusing on the geometry and attributes aspects of GeMS.

[slide: this project] The goal was to not only convert one map but to be able to apply the schema consistently across several maps, and to do it efficiently, avoiding repetitive work by processing multiple maps at once when possible.
And also to establish our own conventions for how to follow GeMS because

[slide: flexible] GeMS is flexible. *Give two of us the same initial dataset and the same GeMS documentation, there might be a dozen different decisions that you and I can make differently.* Establishing *our* conventions for following GeMS allows us to enforce consistency across our data products.

[slide: straight arrow] We would move one or two maps through a series of a few steps, then improve the efficiency of that stage with the goal of processing several maps automatically.

[slide: squiggle arrow] We had to go back and revisit things again because a later step would prompt us to revisit an earlier decision.

[slide: two maps with squiggle arrows] This is iterative in two dimensions. We also had to reevaluate based on the other maps.

The Workflow

[slide: workflow] So I will cover some of those decision points during this very brief overview of the workflow.

[slide: folders] For an individual map, We started with the vector data and FGDC metadata.
We staged three folders: Keep the initial data; you'll probably have to reference it at some point.
We made a copy that we would manipulate before loading it into the GeMS version.
First, we convert the vectors from interchange files to coverage to geodatabase.

[slide: examine data] We can use python to extract unit descriptions from the original FGDC metadata. We used Jupyter notebooks to develop the scripts.

[slide: data source] We would write out the data source that we will paste into the database later. This is an easy thing, and you should do it once and get it right to avoid re-doing it. Note that we chose not to use "DAS1, 2, 3" as the ID; instead we used descriptive and readable source IDs because we found it easier if the data source was immediately obvious from the data source id in other tables.

[slide: workflow split] At this point, you can begin to work in parallel on different parts of the dataset. We pulled every type value, usually this means legend text, into a master glossary

[slide: glossary table] so we can have a broad view of all the different type values across all maps. We also captured all their symbols in this spreadsheet which helped us later.

[slide: striations] with geologist permission, we did a bit of combining terms where appropriate, trying to balance the original wording against the possibility of putting these maps together in the future. You can't always use original wording because the GeMS type field specifies a concise term and shouldn't describe the visual symbol.

[slide: cutbanks] ultimately, we combined some terms but retained differences where they were meaningful. And we also retained the original legend text in an extra column of the feature class.

[slide: workflow, DMU] At the same time, you can build a Description of Map Units table from the legend, and send it to a geologist to complete.

[slide: workflow, sort features] In the meantime, with the feature classes, we decide which GeMS layers are appropriate and separate features into feature classes. We are not in GeMS yet, but we are staging the features by separating contacts from other lines and so on.

[slide: quick reference sheets] to help with this, we made Quick Reference Sheets with GeMS documentation text with one layer per sheet of paper. This helped us focus on the essential reference material. I have the GeMS manuscript in a 3-ring binder, and the quick reference sheets in the front pocket.

[slide: drumlins, striations] This was a decision point where it was beneficial to look at examples of features across many maps. While points could have represented drumlins on one map, lines were more appropriate in another case, so we used lines across the board.

[slide: unique] and sometimes features don't fit a GeMS layer well. If they have a unique set of attributes, it may call for a unique feature class as an extension.

[slide: dir points] We encountered these points that didn't match any prescribed GeMS layer, so we created a new layer following GeMS guidelines.

[slide: workflow, domains] So we have sorted out the destination layers, and we un-do coded domains using a custom ArcToolbox that we created. Then we add fields to match up to GeMS fields and populate them using python at the same time, wherever possible.

[slide: topology] We did a quick topology check to see that contacts cover polygon boundaries. We found several little things to fix like this dashed line that was digitized as individual dashes. At this point, we create a GeMS database and load the features, matching up fields that we've added with the GeMS fields.

[slide: workflow merge] Then we can bring the threads back together again by correcting type values, building the glossary from the master, and importing the DMU.

[slide: GeMS tools] Finally, we start the GeMS checks and validation. We describe any custom fields and layers in the python file so they will be referenced by the metadata script.

[slide: populate] We populate metadata for the feature dataset using ArcCatalog and referencing the resources here. Then we run the metadata tool.

[slide: blank] there are still steps we have yet to implement, such as checking against the original for mistakes and completing symbology. But that's a summary of the workflow so far.

How to do it

[slide: excel] As we followed this workflow, We tracked the progress on an individual map through spreadsheets like this one in Excel.

[slide: trello] And we tracked the progress of all maps through the process using this web tool called Trello, where each map appears like an index card and we advance each map's card from left to right as it moves through the stages.

Concluding thoughts

[slide: progress chart] Here is a chart of our progress on the 21 maps that we set out to convert. I hope this gave you some idea of the workflow, some of our tools and some of our decision points, and the importance of the concept that GeMS is flexible,

[slide: github repo] We have made several of our tools available to you in our github repository, including: Quick reference sheets for the GeMS layers, documentation, python scripts and an Arc Toolbox

[slide: email] We look forward to finishing several of these so that the new data packages will be available and interoperable with other GeMS databases.