Thomas Wright
tgw4@illinois.edu

CS410 Project Documentation and Final Report

Part 1 – Documentation

1. Installation instructions –
This project was quite different from most of the others. It was not about coding, but rather to improve the MeTa and metapy tutorials. Therefore, there is no code to install. Rather, the product of this project was to produce installation instructions for MeTa and metapy for various operating systems. These instructions have been created in the form of github .md files, so they can be transferred to the MeTa and metapy github sites to replace or augment the tutorial instructions already there.

The location of these .md tutorial files are in the directory here:

https://github.com/tgw4uiuc/CourseProject

The files are Meta_tutorial.md and metapy_tutorial.md. Click on either of these files to open them and display the tutorial instructions.

2. Source Code -
This project doesn't have formal source code, per se. That was not the goal of this project, creating tutorial instructions was the goal. Thus, the deliverables are the .md files, which take the place of the 'code' deliverable. "The documentation is the code, and the code is the documentation."

Please note that this project was to improve the existing tutorials, so much of the older tutorial information is still there. For the purposes of the "plagarism" question on part 6 of the grading rubric, I am only claiming the CentOS 8.2.2004, Ubuntu 20.04 LTS, Chromebook and some notes in the general setup sections as my own for the MeTa instructions, and the CentOS 8.2.2004, Ubuntu 20.04 LTS, Chromebook and Windows 10 instructions in the metapy tutorial as my own. The rest of the documentation is from the previous author(s) of the tutorials.

3. Software usage tutorial -
There is a link to the usage video in the video_link.md file in the https://github.com/tgw4uiuc/CourseProject directory. Click that file to see the link to the video.

Part 2 – Project Report

Project Goal:
The goal of this project was to improve the existing tutorials for installing MeTa and metapy on various operating systems. The existing tutorials were outdated and hadn't been updated in several years, and as a consequence, I found many obstacles that had to be overcome to install MeTa and metapy, as

many changes had occurred since then that had prevented easy installation. In most cases, those problems were overcome, and both MeTa and metapy were installed successfully. In addition, since recently produced Chromebooks can now have a beta feature that allows them to use a linux shell, I wanted to investigate the possibility of getting MeTa and metapy working on a Chromebook. This was also achieved successfully.

Project Results:

MeTa was successfully installed on these OSes: (all newer versions of OSes already in the tutorial, or a whole new OS in the case of the Chromebooks):
ChromeOS
Ubuntu 20.04 LTS
CentOS 8.2.2004

Metapy was successfully installed on the above OSes as well, and also confirmed that is it still installable under Win10, with some caveats (python versions 2.7 or 3.4-3.7).

The only one that failed was installing MeTa on Windows 10. I tried the msys2 method as described in the existing tutorial, trying different versions of gcc/g++ and much debugging time, but could not get it to successfully compile. I also tried cygwin, as it is another linux-like environment similar to msys2, but I could not get it to install there either. I had already spent more than half a day trying to get it to work there, and had to give up on that part of the project, as there was no more time for further experimentation there.

Overall, I would say the project was quite successful, and has provided working tutorials for several updated and new OSes that will allow people to get MeTa successfully installed. I would have liked to do tutorials for even more OS versions, but with all of the unexpected problems that were found and the troubleshooting time that was required to find workarounds for them, there was no time to do any others. As it was, just doing these took far more time than the required 20 hours for the project.

Problems Encountered:

There were several problems encountered during the project related to the age of the MeTa and metapy packages and the evolution of newer OS versions since the MeTa and metapy packages were last updated. Those problems are as follows:

1. **gcc/g++ incompatibility** – The way that gcc/g++ handle operator overflow was changed between gcc/g++ versions 7 and 8. This causes problems for the MeTa source code, and it will not successfully compile on versions 8 and above. Thus earlier versions of the compilers must be installed to successfully compile the code.
2. **xlocale.h no longer supported.** This .h file does not exist on many newer linux versions, and thus the MeTa code will error out when trying to compile as it looks for this file. The fix for this is to point xlocale.h to locale.h, which provides the functionality for MeTa to compile successfully.
3. **The icu4c source repository has moved** from icu-project.org to github.com/unicode-org. This causes both the MeTa and metapy code to fail because the make process can no longer download the required source file. The fix for this is to either download the file manually and

copy it into the proper directory in the build files (which is what is done in the tutorials) or to edit the makefiles to point them to the new repository.

4. **Metapy is harder to install on python version 3.8 and higher**, as .whl ('wheel') preprocessed code packages are available for python version 2.7 and 3.4 through 3.7, but not 3.8 and up. These 'wheels' make installation easy, as they do not need to be compiled from scratch, just downloaded and installed. When installing on python 3.8 and higher, they need to be compiled from the source code, which runs into the icu4c repository problem noted above.

Suggestions for improvement of Meta and metapy (and possible project ideas for future students):

There are several things that would greatly improve the usability of MeTa and metapy, especially for new users. These ideas might make good project oportunities for future students to CS410, espeically if they have experience with c++ and makefiles/cmake. These would be addressing the problems noted above.

1. **Update for modern versions of gcc/g++.** Update the code to fix the incompatibilities with versions of the compiler ranging from version 8.0 and up.

2. **Switch the code from using xlocale.h to locale.h.** From what I found researching xlocale vs. locale.h online, there is no functional difference between them for MeTa's needs, but this should of course be confirmed further by the project team.
3. **Point the code/makefiles to the new icu4c source repository.** Change the references from the old icu-project.org repository to the new github.com/unicode-org repository.
4. **For metapy, create new 'wheel' .whl packages for python 3.8 and up.** This will greatly simplify installation for new users.

Time Spent:

Installing SSDs and RAM in test PCs: 1 hour
Downloading install ISOs and creating install USB sticks for several OSes: 1 hour
Installing OSes: 1 hour
Installing MeTa: general installation testing: 1 hour
Installing and Troubleshooting MeTa installation: icu4u repository change: 3 hours
Installing and Troubleshooting MeTa installation: xlocale.h fix: 1 hour
Installing and Troubleshooting MeTa installation: GCC version issues fix: 6 hours
Installing metapy/troubleshooting on Ubuntu: 3 hours
Installing metapy/troubleshooting on CentOS: ½ hour
Investigating which Chromebooks I had access to could support the beta linux mode, and setting that up: 1 hour
Installing and Troubleshooting MeTa on the Chromebook: 4 hours
Installing and Troubleshooting metapy on the Chromebook: 1 hour
Clean re-install of OSes and verifying tutorials work: 5 hours
Writing .md tutorial files: 2 hours

Writing summary report: 2 hours
Writing this final report: 3 hours
Creating, processing and uploading software usage video: 2 hours

Total time spent: 37.5 hours