

Progress Report: Design Updates

Sofya Calvin

December 10, 2016

1 Introduction

1.a AguaClara Overview

AguaClara is an engineering student project team at Cornell University started in 2005 with the goal of bringing sustainable water treatment to the Global South. Through connections with local communities, AguaClara has been able to implement many economic and gravity-powered water treatment plants in Honduras and India. Members on the team carry out research to improve the technologies and then use the results of research for fabrication and generation of AutoCAD designs for water treatment plants and their components. All of this is made possible through the communication between engineers at the water treatment sites and students on campus, which the design team helps facilitate.

1.b Design Team Overview

The Design Team works on various projects which involve automating the design process for AguaClara's water treatment technology. Designs are coded using MathCAD and drawn out via AutoCAD and then published on the AguaClara server for partners to use. The overarching aim of this team is to allow users to enter various parameters about a plant and be presented with detailed AutoCAD designs, a materials list, and a report describing any information needed to build the technology.

2 Low Flow Code Update

As the flow rates entering a plant decreases, the dimensions for the original AguaClara technology become far too small for masons to build and operators to clean. Therefore, as the flow rate decreases, some parts of the plant must be modified and sometimes replaced in order to perform the same task of filtering the water. Currently, the Design team has code that does just this when flow rates dip below 6 Lps. However, in Spring 2016 Paroma Chakravarty's work revealed that it should actually be switched into use beginning at 16 Lps for optimal water filtration as this allows multiple EStARS filters to be used instead of one OStARS filter. Thus, the bulk of this task involves updating the low flow code so that low flow plants are built when flow rates are in the transition phase from *1 Lps to 16 Lps (the reason why the flow rate is cut off at *1 Lps is because this is the cut off rate for using the 1Lps plant). Some things to consider in this task are the optimal number of sedimentation tanks at these low flow rates and the connections between the sedimentation tanks and the EStARS filter as the number of sedimentation tanks changes.

3 Transition to Cloud

The transition to cloud based services is a huge development for the Design team. Cloud computing is a kind of Internet-based computing that allows shared computer processing resources and data to computers and other devices on demand. Essentially, it would provide ease of access to AguaClara designs for global partners as the designs could be made readily available to anyone with access to internet. Furthermore, presenting designs to users on the Cloud using Autodesk's Fusion 360 provides users with the ability to manipulate designs to better understand them. For example, the tool allows users to edit their view of the

design so that they can view the design both from different angles as well as in cross-sectional pieces - a feat which the design team has been trying to accomplish for years.

To implement this transition, the first step is to write new translators to allow already existent MathCAD code to communicate with Fusion 360. Currently, the design team has translators that convert MathCAD code into AutoCAD commands to be inputted into AutoCAD's command line. However, Fusion 360 is a new platform that doesn't have a command line and instead, can be commanded using either Python, C++, or JavaScript. For the design team's purposes, Python seems to be the most viable language for these translators as it is easy to understand and incredibly user friendly. This is especially important because the translators can get to be very complicated.

3.a Details on Using Fusion 360

Almost every feature that partners currently take advantage of using AutoCAD has been integrated into and is accessible on Fusion 360 including creating 2D construction drawings. Because Fusion operates using bodies as objects, however, there is no feature to create layers of the model. When a user downloads the software, they can collaborate on designs in real time by changing them, writing comments on them, and chatting about it while connected to the internet. Since it is not always feasible for engineers on site to be running the program using internet, there is an offline mode that they can make use of which allows them make all of the edits and comments that they normally would, only they wouldn't be updated in real time for other users accessing the same design to see. This likely isn't a problem seeing as at this point when the engineers are actually on site, they are communicating directly about the design in person. For partners without the Fusion 360 software, designs can be accessed on the internet through Fusion's A360. A360 is a web-based site, run and maintained by AutoDesk, where people can view designs from many different views (including section cuts) and comment on them, however they cannot edit the designs.

Other interesting and potentially useful features of Fusion 360 include the functionality to fabricate designs by generating machine code to cut, turn, and mill designs (CAM) as well as show simulations. The former feature, while not exactly applicable now, could be very helpful in the future as AguaClara expands. The latter feature is incredibly helpful in showing how the technology works by simulating the flow of water throughout plants. Not only would this be beneficial for students learning about the technology, but it would also be helpful in pitching AguaClara technology to other organizations in other countries as the project expands outside of Honduras and India.

3.b Commands in Fusion 360

Fusion 360 has commands which can create various shapes or manipulate them in various ways depending on the command. Commands are executed in the API (Application Program Interface) which is a set of routines, protocols and tools that are already built into the software and help execute it. Within the API, commands are represented as a CommandDefinition object and are accessed through `UserInterface.commandDefinitions`. While a command is running, the user cannot interact with it. In other words, if a user commanded the program to draw a sphere of radius 2in, the user cannot change the radius while the command is being executed. This is same situation as in AutoCAD where the user has no ability to change the implementation of a plant once the MathCAD translated text has been entered into the command line. In Fusion 360, the user has the ability to create custom commands using add methods in `commandDefinitions.object`. This could be very helpful in creating some parts of the plant if and when there are certain objects in our design, such as an oddly shaped pipe, that would have to otherwise be created manually because the object isn't already defined in the API. Adding new commands is potentially something that could be written into the Python code however, the team will have to research if there is, in fact a need to add new commands. Otherwise, the code will simply access current commands already built into the program.

3.c Coding Using Python

To code in Python, Fusion 360 uses the application Spyder. When using Fusion, a user can write a script to command the design by going to file, scripts and add-ins, create. When a script is created, it is automatically

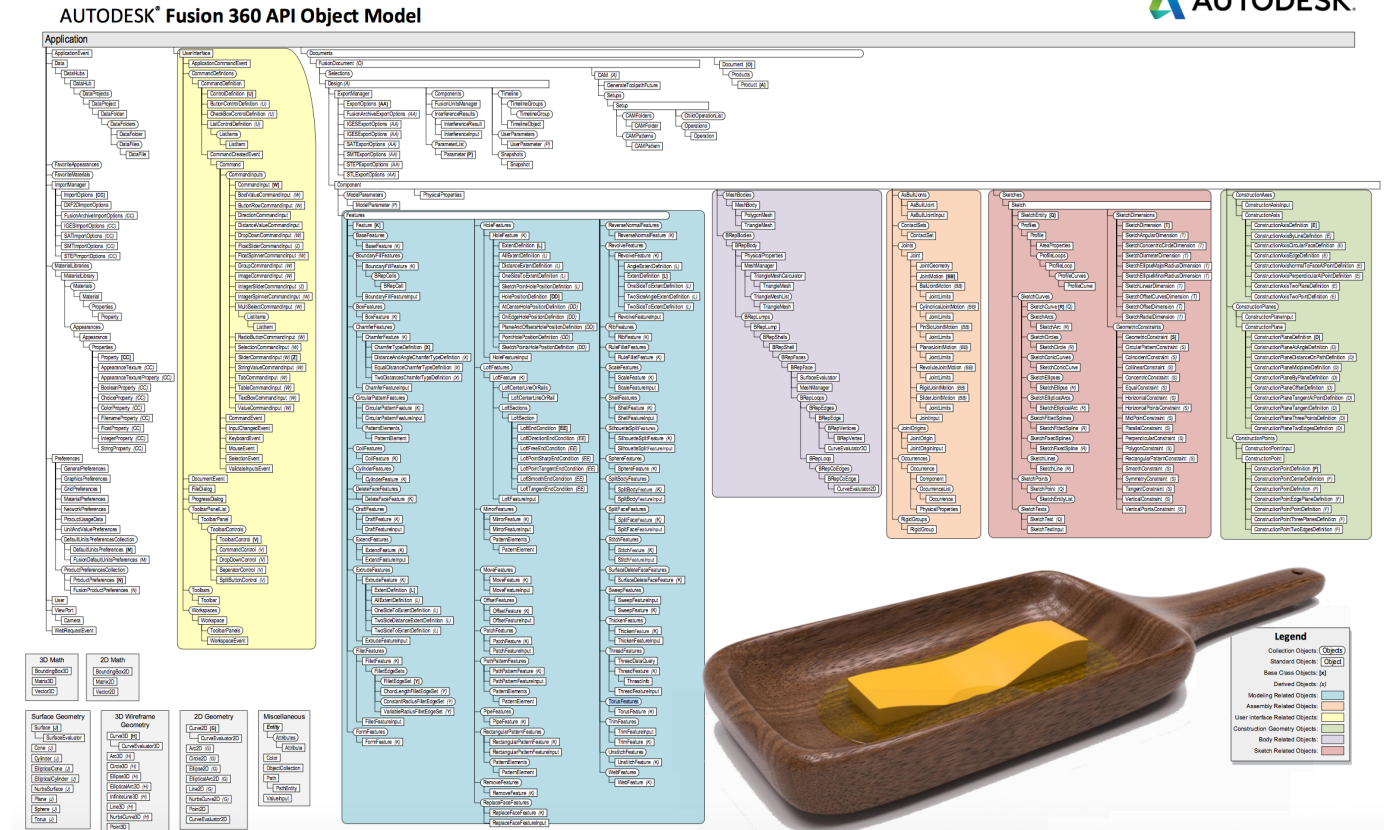


Figure 1: Fusion 360 Object Model

available to edit in the Spyder program, with no downloading of the application necessary. In Spyder, any code written is directly applied to the drawing and can be implemented at any time.

Within Fusion 360, there is a built-in library called `adsk.fusion` which can be accessed when writing python code through Spyder. In it, is the defined object model which is pictured in Figure 1. An incremental factor in writing the translators is going to involve understanding, interpreting, and making use of the Object Model as it outlines various incredibly vital and fundamental objects/classes built into the Fusion 360 software. This is important because any code written in python is going to need to refer to these objects and classes to create bodies. Furthermore, it is likely that many of the components of the plant that are coded for in MathCAD, correspond to certain objects outlined in this model. Thus, in order to properly write the translators, the team is going to have to dive deeper into these objects and see which ones will be necessary to create pipes, tanks, etc.

4 Future work

Moving forward, the team needs to look into the actual syntax of Python and the specific functions necessary to create relevant bodies in the CAD drawing. It seems that much of this information lies within interpreting and understanding the `adsk.fusion` library which is built into the program. CodeAcademy will be a good point of reference for understanding the actual syntax of the Python language. Furthermore, Fusion 360's API Reference Manual has more detailed descriptions of the individual components of the library. From this point, it will just be a matter of taking the MathCAD code and putting it into appropriate Python syntax that can reference the above mentioned objects. This is definitely the most complicated step and will likely require a direct contact with Autodesk, hopefully one at the Ithaca branch, to successfully implement. The Fusion 360 Forum in the Community section of the Autodesk website is also a great resource to help with

specific questions regarding the code.

Finally, the team needs to look into Labview to see how to effectively use it to communicate with Fusion 360 and Spyder. This communication is vital because the server uses Labview to command all of the various programs when a design is requested therefore it is important to make sure that it has the ability to command Fusion 360 in the same way that it commands AutoCAD currently. Information regarding this can likely be found on the National Instruments website or by contacting experts from National Instruments directly.

5 Task Map

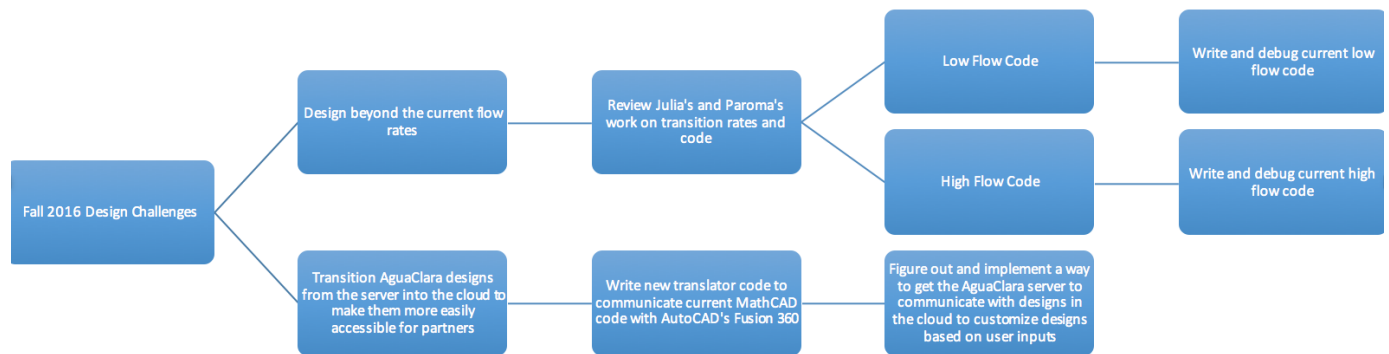


Figure 2: Task Map with challenges

5.a Task Map Details

- Low Flow Rate Plants
 - Review the low flow code already written by Julia Morris in 2012.
 - Read through Paroma Chakravarty’s work from Spring 2016 and figure out to incorporate the results of her research into the current low flow code.
 - Rewrite the parts of the low flow code that need updating.
 - Debug and test updates on the Beta Server.
- Transition to the Cloud
 - Write code that can convert current MathCAD code into commands on Fusion 360 to draw out the plant
 - * Read into the API for Fusion 360 and figure out the best language to use to communicate designs with
 - * Go through current MathCAD to AutoCAD translators and observe how the conversion from language to language works
 - * Write out new translators that convert MathCAD code into a new language that can directly communicate with Fusion 360
 - Figure out a way for designs in Fusion 360’s Cloud to be updated based on user inputs

-
- * Look into how the current system allows users to input parameters and have the server output an appropriate design based on these limitations
 - * Figure out if the current system, which utilizes Labview, can be updated to accomplish this task or if a new program must be used so that user inputs can result in the appropriate designs outputted to the Cloud
 - * Implement this update so that the designs can be fully edited and accessed on the Cloud through Fusion 360