

CS CAPSTONE FALL TERM FINAL PROGRESS REPORT

DECEMBER 6, 2019

AUTOMATE THE SETTINGS THAT CONTROL A MILLION-DOLLAR PRINTING PRESS

PREPARED FOR

HP, INC

PIETER VAN ZEE

RONALD TIPPETTS

PREPARED BY

GROUP62

PROPRIETORS OF THE PRESS

KUAN-YU LAI

COLE JONES

Abstract

This document contains a brief overview of our project, the progress we made throughout the fall term, and our current progress. In the weekly summary, we included the problems we faced every week and the solutions we used to solve those problems. Also, it documents some special things that happened during the term in a figure.

CONTENTS

1	Project Recap	2
2	Current Progress	2
3	Problems	2
4	Week-by-Week Summary	2
4.1	Week 1	2
4.2	Week 2	2
4.3	Week 3	2
4.4	Week 4	2
4.5	Week 5	2
4.6	Week 6	2
4.7	Week 7	3
4.8	Week 8	3
4.9	Week 9	3
4.10	Week 10	3
5	Retrospective	3

1 PROJECT RECAP

The purpose of this project is to simplify the process of choosing the correct press settings for a given print job. Print job settings will be determined through the use of a rules engine, which will take a collection of user inputs about a job (i.e. the type of paper, the selected press, the maximum ink coverage, etc.) and apply them to a set of rules (which may be different for each press). The output of the rules engine will be the optimal press settings for the given print job.

The goal of this project is to reduce the settings selection process to what is essentially a single button click. Users will fill out a form on a website front-end to provide some information about the print job, and the rules engine will do the rest of the work.

2 CURRENT PROGRESS

At this point, our group has created a set of representative rules (essentially a list of factors that affect press settings) to be used for consideration as to the inputs to the rules engine and the composition of the rules that the engine will use. We have also begun to look into two options for use as our rules engine: Drools, an open-source business rules management system created by Red Hat; and Easy-Rules, a lightweight rules engine written in Java. We have not yet decided which engine we want to commit to, so we are doing some testing to find out which engine is easier to use and has better performance.

3 PROBLEMS

We had some difficulty with selecting a platform for hosting our rules engine on a virtual machine. At first we attempted to use Amazon Web Services (AWS), but found that they did not properly limit free accounts. That is to say that they did not restrict free users to free services, and we found that we were being charged for what we thought was free-to-use. In addition, we found that the largest virtual machine size that was available to free users was much too small to host the Drools Docker image (it was a virtual machine with 1GB of RAM). After some trial and error with AWS, we turned to Microsoft Azure, which provides similar services but with a different payment model. Instead of there being some free-to-use services among pay-to-use services, all services are pay-to-use, but free users are given a \$200 credit to use the services as they see fit. With Azure, we are able to host virtual machines with much more RAM that are capable of hosting our rules engine.

4 WEEK-BY-WEEK SUMMARY

4.1 Week 1

Built our resumes and started listing our preferred projects.

4.2 Week 2

Met with our clients and had short tour of HP's Corvallis campus.

4.3 Week 3

Extracted half of the rules from the white papers provided by our clients. Struggled with the requirements from the clients so we had a discussion with our TA.

4.4 Week 4

Finished extracting rules individually and started to merge our documents into one. The time we spent on extracting the rules was a bit longer than we expected, therefore the schedule is a bit behind. We had to spend extra hours during the weekend to catch up.

4.5 Week 5

Finished merging our rules documents and started to attempt to interview professors at OSU. Most of the professors either did not reply to the email. Many of those that did simply said that they did not have time for the interview. We did eventually get one response, and went to his office hours to discuss the project.

4.6 Week 6

Started to do independent research online about potential technologies we could use in the project. The client was fairly vague about what they wanted the project to be, so we didn't really know where to start researching. We resolved this problem by getting some advice from the professor.

4.7 Week 7

Research on the technologies for the project. After a meeting that included the professor we talked to, the client narrowed down the scope of the project.

4.8 Week 8

Research on the web server environment, such as AWS and Microsoft Azure. The free trials of both AWS and Microsoft Azure are fairly limited so we can't have the server up 24/7. As a result, we only launch the online server when doing development or sharing our progress with the clients.

4.9 Week 9

Research on Drools and Docker using the web server. Drools is a bit complicated for what we need, so we looked into alternative rules engines that are simpler. We experimented with other tools such as J-Easy to compare performance and overhead.

4.10 Week 10

Experimental implementation on J-Easy. J-Easy is much easier to learn and use but we still have to compare other factors in order to choose the best tool.

5 RETROSPECTIVE

Positive	Delta	Actions
Professor Scott joined one of our weekly meeting to help clarify the requirements with the clients.	Look into alternatives to Drools, like J-Easy.	Spend time learning Drools and J-Easy so we can create tests to compare overall performance.
Finding Microsoft Azure is a better option than AWS.		Build sample project with Drools so we can compare its performance with J-Easy's.
Finding rules engine (J-Easy) that is simpler than Drools.		