CS Capstone Requirement Document
OCTOBER 25, 2019

Automate the Settings that Control a Million-Dollar Printing Press

PREPARED FOR

HP, Inc

PIETER VAN ZEE

PREPARED BY

Group62

Proprietors of the Press

KUAN-YU LAI
COLE JONES

Abstract

This document describes the details of the automated setting select project that will control the million-dollar presses from HP. It contains subjects like timeline throughout the term, use cases, etc... The main purpose of this document is for people to know what happened during the development process and give the general idea of the final product.

CONTENTS

LIST OF FIGURES

1 OVERVIEW

The objective of this project is to create a decision-making engine that takes an input of information about the PDF file to be printed and a data file with additional information about the job (like paper type, color profile, etc.), and outputs a selection of settings used to control the printing press. The engine will run alongside pre-existing internal tools to analyze print jobs within a queue and generate a settings profile (or choose a pre-existing settings profile) for each job on its way out of the queue into the printing press. Once the printing job has reached the printing press, it will display to the operator the settings it has generated, along with a justification as to why that setting was picked and if it aligned with the settings that were chosen or imported. The overall goal is to streamline the processing of picking the correct settings for a print job, reducing the amount of interaction between the operator and the printing press.

2 GLOSSARY OF TERMS

Term	Description
Ruleset	A set of rules which the decision-making engine uses to generate optimal job settings. Includes information about best practices, optimal ink coverage
Decision-Making Engine	An engine that allows developers build and run machine reasoning models
GUI	Graphical User Interface. The web application used to interact with the engine
API	A protocol that allows client communicate with the server easily

3 USE CASES

3.1 Implementation of Decision-Making Engine

Scenario: A decision-making engine will be implemented. This engine will take a set of inputs and produce a set of press settings to append to a print job.

How: A collection of candidate engines will be selected through interviews with machine-reasoning experts at OSU in addition to independent research. These decision-making engines will have multiple representative rulesets implemented for testing. Once the engine most suited to the task is found, it will be used to create the final implementation.

Success: The final implementation of the engine will be chosen. It will have, in addition to a GUI, an API framework that can be used to communicate with the engine outside of the application.

3.2 Use of Decision-Making Engine in Pre-Press Check

Scenario: The decision-making engine will be used to check if the settings chosen for a job are appropriate.

How: This use of the engine will be part of the pre-press check, a check that occurs before a job is queued to make sure it is fit for printing. The engine will analyze the info about the PDF and applied settings and determine whether or not the job should be printed, and if the chosen settings are appropriate or not.

Success: The job is checked and a justification for why the settings are appropriate or not is produced. The job will not be sent to a queue.

3.3   Use of Decision-Making Engine to Generate Job Settings for Job Ticket

Scenario: The decision-making engine is used to generate optimal job settings from the provided job information. A list of all processed jobs and their outputs is available to the user.

How: The engine is used to process multiple jobs and provide an output that consists of a collection of jobs settings stored in a job ticket in addition to justifications as to why those settings were chosen. The engine is not hooked up to a queue. Instead, the user is able to view a list of completed jobs and their outputs, and can take the generated job settings and save them for later use.

Success: The user will be able to save created optimal job settings in a database, then use them in later jobs that have similar inputs.

3.4   Use of Decision-Displaying Current Operating Task and Progress

Scenario: When the user decides to process their task, they will submit the job ticket so the system can schedule the operation.

How: User submits their job ticket, the ticket will be put in the task queue and wait for the press to process the task.

Success: The user will be able to see all the tasks, job ticket, in task queue and view the status of each tasks.

3.5   Use of GUI to Queue Jobs

Scenario: User has access to a responsive web-based application and make the printing request for the press.

How: The web-based application will have 3 functions: job acquisition, job processing, and job results. Job acquisition only gets the input from the user, the PDF info and job ticket data. The job processing function allows the user to choose some of the printing settings or a settings profile. The settings are different from those that the operator sees, it will be slightly more complicated, with additional information to select rules that the decision-making engine will use. The job results present the current working process of each task and the failure reason if the task failed. Also, it provides a justification for the setting, explaining if the chosen setting matches the setting generated by the decision-making engine or not.

Success: Different users will be able to access the same web-based application to queue up jobs or view the queue. The user will not require press operation expertise due to the user-friendly UI.

3.6   Use of API Calls to Queue Jobs

Scenario: User can directly call the decision-making engine's API to post PDF info and job ticket data to create a print job.

How: A user will use the decision-making engine's built-in API to make a POST call, providing documents relevant to the print job. Similarly to how the hotfolder works, the engine then takes the provided files as input, generate the appropriate settings, and move the job into the print job queue.

Success: The files have successfully been processed by the engine and placed into the print job queue with relevant settings information attached.

3.7   Selection of Rules for Engine

Scenario: A user will be able to specify which ruleset the decision-making engine uses for a particular print job.

How: During the use of the GUI to select appropriate settings for the print job, the user may specify what ruleset the decision-making engine will use. Each ruleset is like a profile that's adapted to fit certain constraints better, such as the type of paper or the printing press being used.

Success: The settings created for a print job will be better-suited for the type of paper and press being used depending on the ruleset used.

4   TOOLS & APPLICATIONS

4.1   Existing Internal Tool

There exists an internal tool within HP that allows users to import a PDF of what they want to be printed and XML file that contains some rules about the print job (such as width, paper type, etc.). The user can then select a myriad of settings that the printing press will use to print the job. We will be piggybacking on this tool, inserting our decision-making engine into the pipeline between this internal tool and the print job queue.

4.2   Decision-Making Engine

Our decision-making engine will take the settings chosen from the internal tool (or from an XML file) in addition to the PDF of what to print and generate what it determines to be optimal settings for the job. The settings it chooses are based on its analysis of the PDF info and job ticket data. It analyses the size of the document, the density of ink per page, the use of gradients, color fills, and graphics, and determines the optimal printing speed, press tension, and drying heat/time. It will attach the generated settings to the print job when it adds it to the queue, in addition to supplying a justification for each setting choice and a log file to analysis.

4.3   GUI

The GUI will be created for user-friendly interaction with the decision-making engine. It will take a PDF info and job ticket data as input. The site will allow the user to drag and drop files into a web interface. It will also allow users to select which ruleset they want the engine to use.

4.4   API

An API will be created for the engine to permit its use without having to interact with the GUI. It will be necessary for other applications to provide input without human interference, permitting the engine to run alongside existing programs in the job creation pipeline.

4.5   Database

An external database will exist to hold all of the generated rulesets that the engine will use. It will also hold engine output that is deemed reusable for jobs with similar inputs.
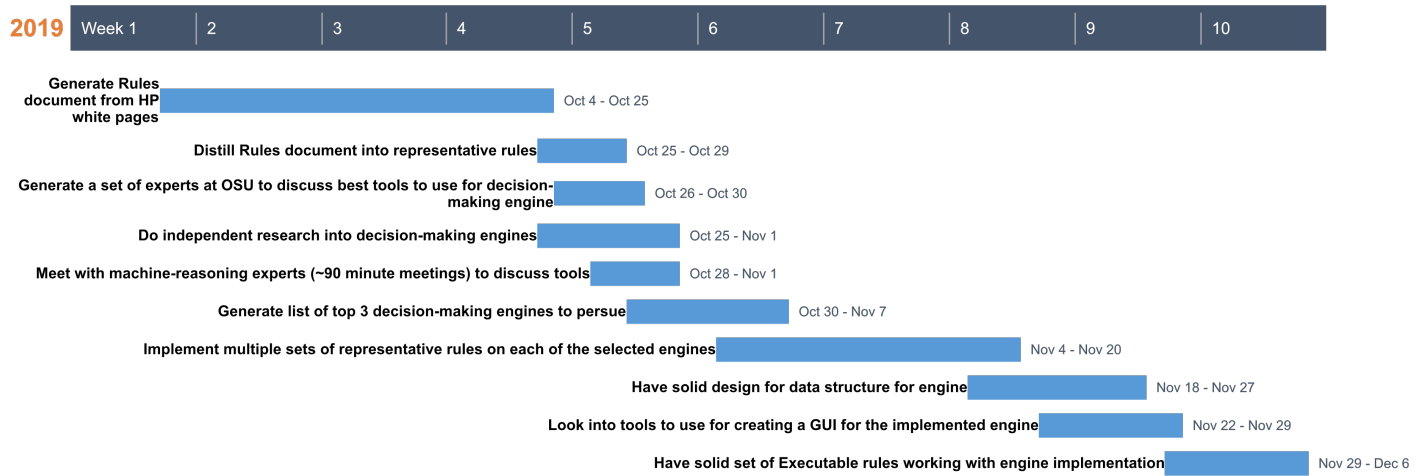
5 GANTT CHART

2019	Week 1	2	3	4	5	6	7	8	9	10

Generate Rules document from HP white pages — Oct 4 - Oct 25

Distill Rules document into representative rules — Oct 25 - Oct 29

Generate a set of experts at OSU to discuss best tools to use for decision-making engine — Oct 26 - Oct 30

Do independent research into decision-making engines — Oct 25 - Nov 1

Meet with machine-reasoning experts (~90 minute meetings) to discuss tools — Oct 28 - Nov 1

Generate list of top 3 decision-making engines to persue — Oct 30 - Nov 7

Implement multiple sets of representative rules on each of the selected engines — Nov 4 - Nov 20

Have solid design for data structure for engine — Nov 18 - Nov 27

Look into tools to use for creating a GUI for the implemented engine — Nov 22 - Nov 29

Have solid set of Executable rules working with engine implementation — Nov 29 - Dec 6

Fig. 1. Gantt Chart