CS CAPSTONE REQUIREMENTS DOCUMENT
NOVEMBER 29, 2018

STRATEGIC PLANNING IN EMERGENCY SERVICES

PREPARED FOR

LEVRUM

CARL NIEDNER

PREPARED BY

GROUP 66

CODE7

ETHAN AHUJA
LAUREN GASTINEAU
DAVID SAHNI
MAXIMILIAN VAN HOUT
TRENT VASQUEZ

Abstract

As a client of Levrum, we will help them branch out and develop a piece of software to aid in hospital emergency room management. This piece of software is not intended to be a final product or extension of their current software suite, but its own dive into experimental software that Levrum may expand upon it in the future. The software will be broken into the following segments: hospital predictive software, a hospital simulator, a multi-objective search algorithm for hospital input, and the recommendation output user interface. To satisfy Leverm′s goals for the project, each of these components needs to work together to produce meaningful predictions for hospital staff.

CONTENTS

1 INTRODUCTION

In the following document, the design goals and components will be highlighted in great detail as to provide a framework in which they will be developed. Our purpose will cover the desired input and output of the system whereas the scope will cover more specific requirements. The glossary and references will help define the terminology that will be covered later in the document as well as references to our research on the technology and implementations. Our overview will summarize the nature of the system and the components that will be further detailed in section two of the document. Following the content of the overview, a detailed analysis of the requirements and their intended implementation will be laid out. These implementations will include the technologies intended for use as well as the desired inputs and outputs of each segment.

1.1 Purpose

The purpose of the document is to highlight the design goals set forth by Levrum the methods in which we will develop them. The system that is currently proposed is a simulator-optimizer-recommendation engine tailored to the needs of an emergency room (ER). The document will break down the requirements of each of those components and define what their role is in the system. In addition to defining their role, we will also discuss the desired inputs and outputs for each component. While Levrum does not expect the final state of our project to be marketable, the goal is to have a system in which the final output is within the current scope of the professional medical field.

1.2 Scope

This document gives a detailed overview of our project, its functional, and its performance requirements. However, the methods on which we will be fulfilling these requirements is not included. This document contains an abstract overview of our project's broad specifications including system interfaces with other software and the functional and performance requirements our project will fulfill. Furthermore, this document details the functions, assumptions, and constraints our project will operate within.

1.3 Glossary

- **ER:** An Emergency Room is the department of a hospital that provides immediate treatment for acute illnesses and trauma [4]. The software package we will be producing that helps to increase the performance of ER's
- **Monte Carlo:** The Monte Carlo method is a technique for risk analysis in simulations that work "by building models of possible results by substituting a range of valuesa probability distributionfor any factor that has inherent uncertainty" [1]. The simulator will receive numerous sets of data chosen by the Monte Carlo method at random that will be plotted as a distribution of expected values.
- **Discreet Event Simulation:** A discrete event simulation is a modeling technique where "patterns of events in the problem are recreated so that the timing and resource implications can be examined" [5]. Each event affects the current state of the simulation at a specific time.
- **Simulator:** The program that will carry out an accurate representation of an ER to be improved upon by the Optimizer. This will output the metric values. It will be a discrete event simulator.
- **Optimizer:** The machine learning algorithm that influences the simulation. When analyzing the produced metric data from the previous iteration of the simulation, the Optimizer can making configuration changes that will yield more positive results.

- **Predictor:** The Predictor is a program that analyzes the historical data from an ER. The program looks for trends in the data that can be used to produce ranges for the event's an ER will have to complete.
- **Metrics:** The output from the simulation that includes an analysis of how well the simulation performed given a set of parameters. Metric data is used by the Optimizer as insights as to what can be changed to improve performance.
- **Historic data:** The anonymous data provided by an ER that is used to predict trends that the ER will face now and in the future.
- **Recommendations:** Recommendations are the final reports produced to the user that contain probability analysis graphs that tell the user what to expect in the future and what the best way to plan is.
- **API:** - An Application Programming Interface is a "set of commands, functions, protocols, and objects that programmers can use to create software or interact with an external system" [2].
- **GUI:** - A Graphical User Interface, sometimes referred to as a $gooey$, is the graphical interface that a user interacts with that improves the users experience over that of a command line [3].
- **Runtime:** - The runtime used to define performance requirements QR4, QR5, QR5 is measured in real time, the actual amount of seconds that pass. It not an analysis of the algorithms asymptotic runtime

1.4 References

1 Monte Carlo Simulation: What Is It and How Does It Work? - Palisade. (2018). What is Monte Carlo Simulation?. [online] Available at: http://www.palisade.com/risk/monte_carlo_simulation.asp [Accessed 28 Nov. 2018].

2 Techterms.com. (2018). API (Application Program Interface) Definition. [online] Available at: https://techterms.com/definition/api [Accessed 28 Nov. 2018].

3 Techterms.com. (2018). GUI (Graphical User Interface) Definition. [online] Available at https://techterms.com/definition/gui [Accessed 28 Nov. 2018].

4 Merriam-webster.com. (2018). Definition of EMERGENCY ROOM. [online] Available at: https://www.merriam-webster.com/dictionary/emergency%20room [Accessed 28 Nov. 2018].

5 IGI-Global. (2018). What is Discrete Event Simulation. [online] Available at: https://www.igi-global.com/dictionary/discrete-event-simulation/7878 [Accessed 29 Nov. 2018].

1.5 Overview

The remaining sections of this document will detail an overall project description and specific project requirements. In this document's second section, overall description, our project is described in more detail, including its functions, constraints, dependencies, and attributes. The types of users and how they will use our system is also discussed. The third section of this document gives a detailed explanation of all functional and performance requirements this project will meet. Each functional requirement will describe what each system component will provide for the user, the abilities of each component, and what the component depends on. Each performance requirement will detail what system metric is being measured, how each metric will be measured, and our goals for each measurement outcome.

2 Overall Description

This second section of our document, overview description, will give a detailed overview of our project. This section covers what our software will provide, its main functions, constraints, assumptions and dependencies, and software attributes. The types of end users and how they will interact with our system is also discussed. The end of this section is a gantt chart that outlines our project timeline on a task-by-task basis. This gantt chart will start in fall term and will end in spring term.

2.1 Product Perspective

The software we are building is a system that is composed of five major components: a predictor, simulator, optimizer, resiliency control, and data analyzer. This system will take in historic ER data as its primary input, and our predictor will output predictive future ER datasets. These datasets interface with a discrete event simulator and our simulator will create models of the ER. Each model will recommend different objectives with different trade off values. The optimizer will interface with the simulator and will build an ER model. The optimizer will construct progressively better sets of recommendations. To provide resiliency control a monte carlo algorithm will generate variations of the data such that a given recommendation can be tested for resilience. We will also develop a GUI to take in the user ER problem constraints for our simulator to develop ER models and we will develop the functionality to display the recommendations and their metrics.

2.2 Product Functions

Our ER simulator will accept constraint input from the user through an input GUI. With this constraint input GUI, the user will be able to generate a configuration (config) file that details the constraints of the ER. This config file will be used to set up the simulator and optimizer. When the system is finished, the user will be presented with a GUI that displays the final recommendations that a user is advised to make to their ER and the associated metrics. Some example final recommendations includes methods to minimize ER operational costs, maximize patient outcomes, and minimize waiting times.

2.3 User Characteristics

The primary users for this software will be hospital administrators and staff who are looking for decision support regarding emergency room equipment, staffing, and future planning. They will have some technical knowledge and will have strong domain knowledge regarding hospital systems. They will be looking for different sets of recommendations with different tradeoffs to evaluate. They should be able to configure the simulation model with their specific emergency department constraints. The system will provide data visualization graphs to support its recommendations. There is a specific set of users who will interface with the software, so there is no real need to treat users differently.

2.4 Constraints

The primary constraint for this project is the large ER search domain we are approaching. Searching this space will require many iterations of simulation, and therefore our simulator must operate in a reasonable timespan. An additional constraint on the simulator is that there must be APIs to programatically update an ER simulation model to enable the optimizer to perform its task.

2.5 Assumptions and Dependencies

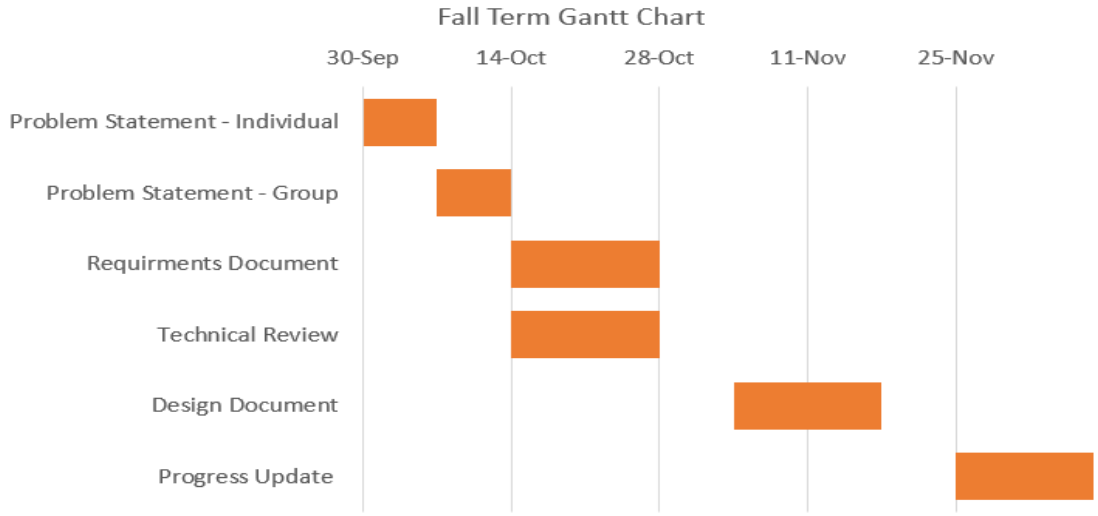
There are a number of dependencies associated with this project, the first being the availability of historical ER data. Historical emergency arrival data will be ideal for testing the simulator and optimizer, however data can be created if needed. When the end user has the product, it will be dependent on their ability to supply hospital specific historic data. Levrum will assist us in obtaining data. Additionally, we have a dependency on the simulator we choose to use. The simulator must have functionality that allows the optimizer to update the model, run the simulation, and analyze results.

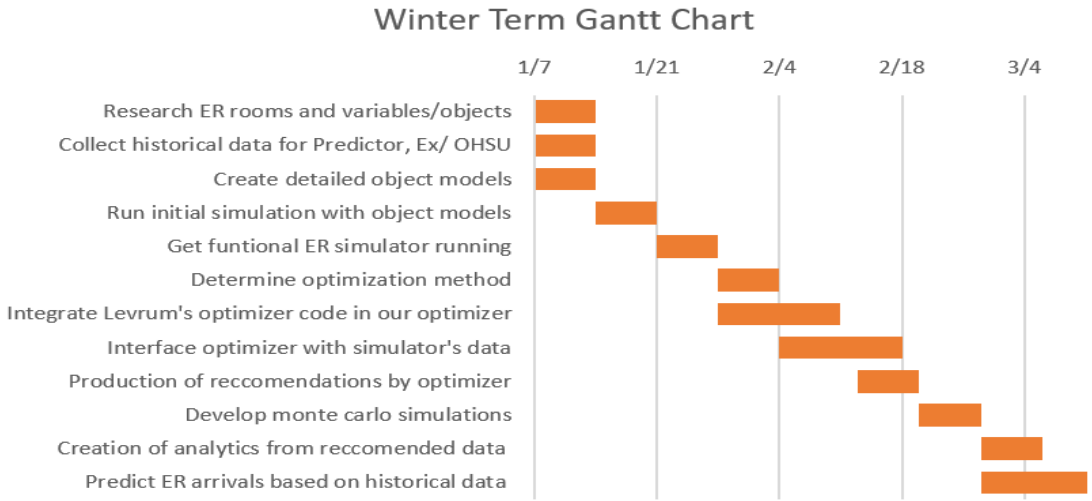
2.6 Software Attributes

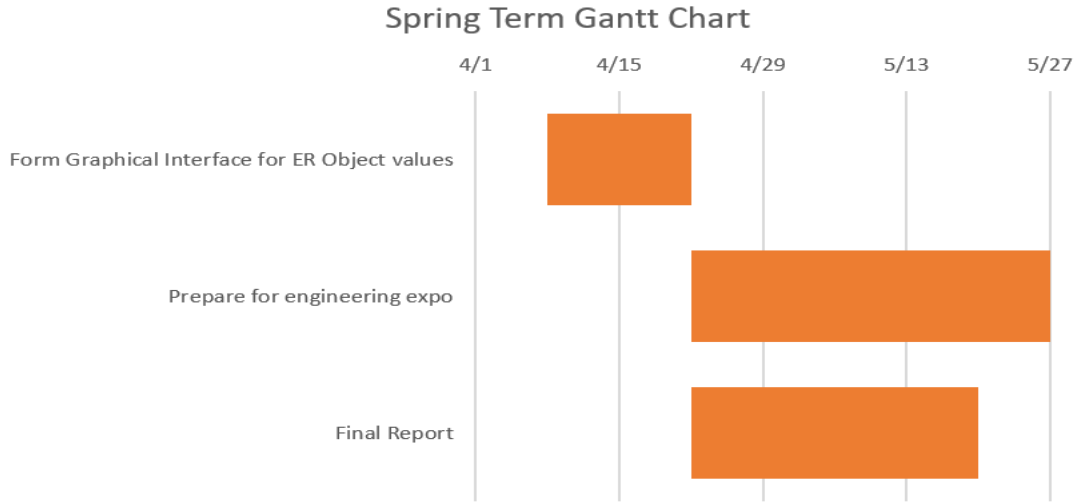
The goal of this project is to develop a prototype software suite that performs data prediction, discrete simulation, and mutli-objective search in the emergency department domain. With that in mind, other attributes like applicability and security are less important than completing a first working version of the product. One key software attribute will be maintainability, as the project must be well documented to allow for future development by Levrum.

2.7 Appointing of Requirements

These Gantt charts outline our fall, winter, and spring term project planning by week. Our project is planned by a task-by-task basis. These Gantt charts start in fall term's week 3 and end in spring term's week 10.







3 SPECIFIC REQUIREMENTS

This section contains specific functional and performance requirements of our project. Thus, this section outlines what our system will do and how our system will work. Each functional requirement will describe each system component. The abilities of each component, what the component is dependent upon, and how this component will benefit a user is provided. Following this, each performance requirement is described. These requirements cover which system metric is being measured, the methods that will be used to measure each metric, and our goals for each measurement outcome.

3.1 Interfaces

3.1.1 User Interfaces

The users of our system have minimal required interactions. Users will initially boot up the program in a command line. A GUI will appears with fields for users to enter their specific ER constraints. This generates a config file for the simulator. The user is then told the ER simulator is running. When the simulation is finished, the user will be presented with different recommendations for their ER and graphs / charts of their results.

3.1.2 Software Interfaces

There are essentially 4 stages of software that need to interface. First historic data must be fed into the predictor, which will pass data into a monte carlo simulator to create events. These events will be used as input for the emergency department discrete event simulator. The ER simulator will interface with the search algorithm, or optimizer, to run many simulations analyzing the simulation output and adjusting the simulation model. The interfacing of data between these four portions is a major requirement for the system.

3.2 Functional Requirements

This section defines all functions of our system. A function is described as a specific behavior between inputs and outputs.

ID: FR1
TITLE: Load historical data
DESC: User should be able to load historic data into predictor
RAT: To predict new data, the user needs to load old data
DEPEND: None

ID: FR2
TITLE: Generate Predicted data
DESC: Predictor should generate new data from historic data
RAT: To provide accurate future models, prediction is necessary
DEPEND: FR1

ID: FR3
TITLE: Create Simulation Model

DESC: Users should be able to create representation of their ER in a model
RAT: To run accurate simulations, users need accurate models of their ER's
DEPEND: None

ID: FR4
TITLE: Generate Simulation Events
DESC: Use predicted data to generate events to occur in simulation
RAT: Simulate ER operation using these events created from predicted data
DEPEND: FR2

ID: FR5
TITLE: Perform Simulation
DESC: Run a simulation with a given hospital model and event set
RAT: To run our optimizer, we will need to successfully run simulations
DEPEND: FR4, FR3

ID: FR6
TITLE: Programmatically run simulation
DESC: Be able to take a simulation model and events and start a simulation from a program
RAT: For the optimizer to run many simulations it, it must be able to start them programmatically
DEPEND: FR5

ID: FR7
TITLE: Perform multi objective search on known problem
DESC: Implement search algorithm in language of choice to solve known problem set
RAT: Test the algorithm on known problems to ensure it works
DEPEND: None

ID: FR8
TITLE: Perform Optimization on simulation results
DESC: Use the output of the simulation to and the search algorithm to optimize simulation
RAT: To create recommendations must optimize towards optimal results
DEPEND: FR7, FR6

ID: FR9
TITLE: Create Recommendation
DESC: Use optimized simulation models to create recommendations to users
RAT: The core deliverable, the ER model recommendation to the user
DEPEND: FR8

ID: FR10

TITLE: Provide Data visualization

DESC: Use data visualization software to show how the optimizer came to its recommendation

RAT: Provide support for the recommendation given

DEPEND: FR9

ID: FR11

TITLE: Provide GUI to system

DESC: Create a simple gui to load in models/configurations and start process from

RAT: Provide visual feedback for users/demos

DEPEND: None

ID: FR12

TITLE: Resiliency Data generation

DESC: Use monte carlo simulations to create data events to test model resiliency

RAT: Important portion of any simulations is to ensure the model is resilient

DEPEND: FR2

3.3 Performance Requirements

At this time, there are no current performance requirements defined by our client.