2015

An Integrated Framework for Automated Data Collection and Processing for Discrete Event Simulation Models

Carlos Rodriguez
University of Central Florida

AN INTEGRATED FRAMEWORK FOR AUTOMATED DATA COLLECTION AND PROCESSING FOR DISCRETE EVENT SIMULATION MODELS

by

CARLOS MANUEL RODRIGUEZ
B.S. University of Minnesota Twin Cities, 2001
B.S.B. University of Minnesota Twin Cities, 2002
M.S.P.E. University of Illinois Urbana-Champaign, 2004
M.S. University of Illinois Urbana-Champaign, 2005
M.A. Boston University, 2008

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2015

Major Professor: J. Peter Kincaid

ABSTRACT

Discrete Events Simulation (DES) is a powerful tool of modeling and analysis used in different disciplines. DES models require data in order to determine the different parameters that drive the simulations. The literature about DES input data management indicates that the preparation of necessary input data is often a highly manual process, which causes inefficiencies, significant time consumption and a negative user experience.

The focus of this research investigation is addressing the manual data collection and processing (MDCAP) problem prevalent in DES projects. This research investigation presents an integrated framework to solve the MDCAP problem by classifying the data needed for DES projects into three generic classes. Such classification permits automating and streamlining the preparation of the data, allowing DES modelers to collect, update, visualize, fit, validate, tally and test data in real-time, by performing intuitive actions. In addition to the proposed theoretical framework, this project introduces an innovative user interface that was programmed based on the ideas of the proposed framework. The interface is called DESI, which stands for Discrete Event Simulation Inputs.

The proposed integrated framework to automate DES input data preparation was evaluated against benchmark measures presented in the literature in order to show its positive impact in DES input data management. This research investigation demonstrates that the proposed framework, instantiated by the DESI interface, addresses current gaps in the field, reduces the time devoted to input data management within DES projects and advances the state-of-the-art in DES input data management automation.

Dedico este esfuerzo a mis maravillosos padres Belinda y Antonio, a quienes todo les debo y cuyo amor, esfuerzo, consejo y ejemplo me han permitido llegar hasta este punto de mi vida. Que este esfuerzo sirva igualmente de ejemplo a mi hijo Ian Emiliano, a quien amo por sobre todas las cosas en la vida. Todo lo puedo en Cristo que me fortalece. En El confio.

ACKNOWLEDGMENTS

I would like to thank Dr. Christopher Geiger for helping me organize my thoughts in the proper manner and focus my efforts in the right direction. I would like to thank Dr. John Peter Kincaid for his support and constant willingness to help.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

AD	Anderson-Darling goodness-of-fit test
DES	Discrete event simulation
DESI	Discrete event simulation inputs
GDM	Generic data management
GOF	Goodness-of-fit
IAT	Prefix for MySQL table names for Class (I) data
IID	Independent and identically-distributed
KS	Kolmogorov-Smirnov goodness-of-fit test
MDA	Machine data acquisition
MDCAP	Manual data collection and processing
OU	Prefix for MySQL table names for Class (III) data
PRPC	Press to resume a previous collection
SQL	Structured query language
ST	Prefix for MySQL table names for Class (II) data
VN	Variable name

CHAPTER ONE: INTRODUCTION

The Importance of Data Collection in Discrete Event Simulation

Discrete Event Simulation (DES) is widely regarded as a very powerful tool of modeling and analysis. DES is used in different disciplines, and its usage will continue to grow [1], fueled to a large extent, by the continuous enhancements in computer capabilities [3]. Areas of application of DES include, but are not limited to manufacturing, health care, logistics, military applications, finance and business, academia, human resources and energy [8]. Within these areas, DES has had an important impact in design, planning and control, strategy making, resource allocation and training [6].

DES models rely heavily on data in order to estimate different parameters that drive the models through simulated time [3]. It is universally accepted that quality data are key drivers of competitive advantage and proper decision making [32]. Such convention applies directly to DES modeling, where results are sensitive to the quality of the data used to build the simulation models.

The goal of this research project is developing the foundations for programming a robust automated alternative to replace manual data collection and processing (MDCAP). As is detailed later in this document, MDCAP is a common practice among DES researchers and practitioners, most of whom agree that more automation in the collection and processing of DES input data is highly desirable [8].

Motivation

Robertson and Perera pointed out that “data collection is an extremely time consuming process predominantly because the task is manually orientated.” [7]. This is especially the case when the data are collected through personal observation such as through-time studies or collected via database queries of historical processes or system performance.

The challenges of MDCAP are not so prevalent when obtaining DES model input data from system or process documentation such as process plans, facility layouts and work procedures, from personal interviews with subject matter experts, from vendor claims, from design estimates and specifications, and from the open research literature, which are all common sources of data to construct DES models. In the framework presented in this research investigation, such valid sources of data are used to build the model logic. As will be discussed later in this document, the model logic has a very specific role in the development of a new approach to automate data collection and processing for DES models.

Further manual processing of collected data is often required beyond the computations performed in a spreadsheet because the data sometimes need to be converted into a format that is compatible with a probability distribution fitting software, such as ExpertFit® by Averill M. Law and Associates, Inc., EasyFit® by MathWave Technologies, Stat::Fit® by Geer Mountain Software, Minitab® by Minitab, Inc., and R, a free software programming language.

Therefore, the focus of this research is when the simulation modeler or analyst is tasked with the collection of historical performance data from a stochastic, dynamic process or system.

Some of the simulation researcher - and practitioner - recognized challenges of MDCAP include:

- Manual data collection including data entry, which increases the likelihood of data entry errors arising from human manipulation of data;
- Data pre-processing in the form of reformatting and calculations, e.g., data preparation for probability distribution fitting, the calculation of interarrival time parameters, service time parameters and historical probability parameters, etc.; and
- Multiple file handing in order to maintain and process data, which can become particularly burdensome psychologically.

This research proposes a framework to eliminate the undesirable characteristics of MDCAP using a robust, innovative, intuitive, automated and user-friendly solution that can be used across the different disciplines where DES modeling is appropriate.

State-of-the-Art in Automated DES Data Collection and Processing

The state-of-the-art in DES input data management automation includes solutions that are highly customized and specific to the project being carried out. In other words, the state-of-the-art is moving away from generalization and towards project-specific customization. The documented state-of-the-art solutions are exclusive to manufacturing applications of DES and are considered “too difficult” to implement on a consistent basis [8], especially outside of manufacturing. Such difficulties are confirmed by the fact that most DES practitioners still use manual approaches to prepare data for DES models. As of 2012, 80% of practitioners still rely on some type of manual gathering and processing of input data for DES projects [8].

There are documented solutions such as the Generic Data Management Tool (GDM-Tool) [14] and Machine Data Acquisition (MDA) [15], which do address some of the challenges of manual data processing, but they rely on previously available data found within corporate data systems. Such reliance on the existence of previous data imposes challenges to the proliferation of these kinds of solutions because data are often not available or if available, these are often not ready to be used in DES models. This suggests that in order to advance the state-of-the-art, there needs to be less reliance on often-unavailable corporate data systems and more reliance on non-available but collectable data. Furthermore, in order to advance the state-of-the-art, new solutions must be usable beyond manufacturing applications of DES.

Currently, the streamlined real-time integration of data collection, data calculations, probability distribution fitting and assumptions checking does not exist in DES input data management literature. As a matter of fact, there is no documented attempt to accomplish such tasks in a streamlined, automated manner. Furthermore, there is no theoretical foundation that supports the development of such a solution. The fact that there is no integrated solution that starts with the collection of data, leads to isolated processing of several steps within DES input data management, potentially causing undesirable time delays in the completion of DES projects.

A New Alternative for DES Input Data Management

The approach proposed in this research starts with the analyst collecting the raw data based on a well-defined framework by performing actions such as pressing buttons with computer mouse clicks.

This project proposes three general classes of data for DES projects, regardless of the area of application. The intended classification of the data is the foundation of the proposed framework in order to achieve DES data processing automation. In this research, the data that are collected are assumed to be independent and identically-distributed (IID), nevertheless the proposed framework permits the automated validation of such critical assumption [3] in real-time. The three general classes of input data can be conceptualized as follows:

I. The class of data when the ending time of one event is the start time of the next event (e.g., interarrival times).

II. The class of data when the ending time of one event is not the start time of the next event; hence, there must be a start time and an end time for each event (e.g., service times, processing times, production times, transport times, etc.).

III. The class of data when tallying different outcomes at a decision point generates relative frequencies, which represent historical probabilities, where the sum of the probabilities of each distinct outcome at a given decision point equals 1.0 [4].

At least one of these three conceptual data classes is necessary in any DES project [3, 4] in addition to the model logic. Moreover, as the complexity of the subject process or system increases, the likelihood of including data from all three classes increases. This implies that data needed for any DES project fall among or can be derived from these three general data classes. Throughout the remainder of this document, these general data classes are referred to as Class (I), Class (II) and Class (III). Formal definitions of Class (I), Class (II) and Class (III) data are provided in chapter three, where the proposed framework is presented in detail.

Claim 1 Data of Class (I), Class (II) and/or Class (III) suffice to put together any DES project, given the model logic that allows identifying the necessary parameters.

The development of the proposed framework to address MDCAP must rely on **Claim 1**. The existence of a model logic is essential in simulation models; Skoogh and Johansson indicate that “the first step while preparing input data for simulation models is to identify which parameters are necessary to include in the model” and that “all parameters need to be defined with regard to how they shall be measured and represented in the model” [5].

The proposed framework also needs to be robust; that is, it must be applicable across different disciplines that utilize DES. Such framework is the theoretical foundation for programming a computer solution that automates input data management for DES models. If the proposed automated solution can effectively collect Class (I), Class (II) and Class (III) data, then this solution can be used across different disciplines where DES modeling is appropriate. This reasoning leads to the following:

Claim 2 Any solution that can collect Class (I) data, Class (II) data and Class (III) data can be used on any DES project, regardless of the discipline or area of application.

The development of a computer solution to address MDCAP must rely on **Claim 2**. Such automated solution is the instantiation of the proposed integrated framework for automated DES data collection and processing.

Research Objectives

The aim of this research investigation is to produce an innovative solution in the area of data collection, processing and management for DES modeling. The ultimate goal of this investigation is to develop a framework where user inputs automatically generate data collection, processing and management in order to accelerate the construction of DES models with the intent of reducing the life cycle of DES studies. The specific objectives of this research are as follows:

Objective 1: Formulate the MDCAP problem within the context of DES models;

Objective 2: Design an integrated theoretical framework to address the MDCAP problem. The proposed framework will provide the foundations to rapidly process raw data in real-time and generate the necessary data computations, transformations (e.g., distributions, probabilities, etc.), graphics and validations required for DES models. The framework will not only be based on classical statistical modeling principles but it will also be designed using user-centered design and human-computer interaction design principles to enhance the overall user experience.

Objective 3: Compare the proposed integrated framework to other automated solutions for DES input data management such as the Generic Data Management (GDM) Tool and Machine Data Acquisition (MDA) in order to demonstrate that the proposed integrated framework addresses current gaps in the field; and

Objective 4: Perform usability studies in order to measure the impact of the proposed framework and maximize its compatibility with the needs and desired uses of DES modelers and analysts.

Expected Contributions of this Research Investigation

This investigation should contribute quite significantly to the body of knowledge and advance the state-of-the-art in DES data collection and management. Specifically, this investigation adds the following original contributions to the body of knowledge:

i. The conceptualization of data needed for DES projects into three specific data classes, which permits focusing on each data class in order to automate its collection and processing.

ii. A generic approach to automate input data management that is applicable across the different disciplines where DES modeling and analysis are appropriate.

iii. Input data management automation for DES models without relying on previously available data within internal data sources or corporate data systems.

This research will potentially reduce the time devoted to MDCAP because the proposed solution will streamline and automate the collection and processing of the data in real-time in response to user inputs. In addition, automation will potentially reduce data entry errors and increase data quality due to less human intervention. Finally, this research will provide a mechanism for the real-time validation of DES theoretical assumptions, guaranteeing the theoretical validity of the resulting parameters and consequently of the simulation models [3].

Several studies suggest that the amount of time within DES projects devoted to input data management is about 31% of the entire project's time, and could scale up to 40% [5, 33]. The fact that most DES analysts and practitioners desire solutions that automate DES input data management [8] suggests that the proposed solution will be attractive to users due to the avoidance of manual data collection and processing.

Organization of this Document

The remainder of this document is organized as follows. Chapter two presents a review of the existing research literature about the state-of-the-art in automated data collection and processing for DES models. The details about the proposed integrated framework are presented in chapter three. An instantiation of the proposed framework is detailed in chapter four in the form of the DESI user interface. Chapter five presents a comparison between the DESI interface and other documented solutions for DES input data management automation. Chapter six evaluates the impact of the proposed framework on DES projects' time-to-completion. Chapter seven presents suggestions for future research and the conclusion.

CHAPTER TWO: LITERATURE REVIEW

Methodologies for DES Data Collection

The process of data collection, data entry and data management can take considerable amount of time within DES modeling projects. The time spent on data management is, on average, about 31% of the DES project's lifecycle, and can go up to 40% [5, 33]. Different research has been focused on analyzing and improving the process of collecting, entering and managing data for DES projects, nevertheless such efforts are considerably less than the efforts devoted to statistical techniques for data analysis [12]. Simulation practitioners acknowledge that different DES software have limited functionalities for data preparation [12] and usually present insufficient support for input data management [8].

In DES modeling projects, data can be classified based on availability as "non-available and non-collectable", "non-available but collectable" and "available" [9]. "Non-available and non-collectable" data can be approximated by interviewing subject matter experts or by using data from similar systems or by using available data previously measured and stored in process libraries [10].

Data categorized as "available", in DES applications, are usually generated by internal data systems such as Enterprise Resource Planning (ERP) systems, Material Planning Systems (MPS), Manufacturing Execution Systems (MES) and previous analysis efforts [5]. "Available" data are often not ready to be used in simulation models and further processing is required [5].

Depending on where the data are stored, "available" data are often accessible thru customized tools that may or may not be able to prepare the data for DES projects. Such tools are often referred to as ETL (extract, transform, load) applications, which can export, process and move data between different sources [11]. Documented solutions for DES automated input data management rely on the existence of "available" data [14, 15, 29]. Such reliance on "available" data imposes important limitations to the proliferation of these solutions because the implementation of internal data systems can be complex and costly.

"Non-available, but collectable" data for DES projects can be gathered following different approaches or methodologies. The most common approach is to conduct time studies and measure the relevant steps of the subject process or system [5]. Such a manual approach would fall under input data management methodologies (A) and (B) presented by Robertson and Perera [7, 8]. **Figure 1** summarizes the four DES input data management methodologies.

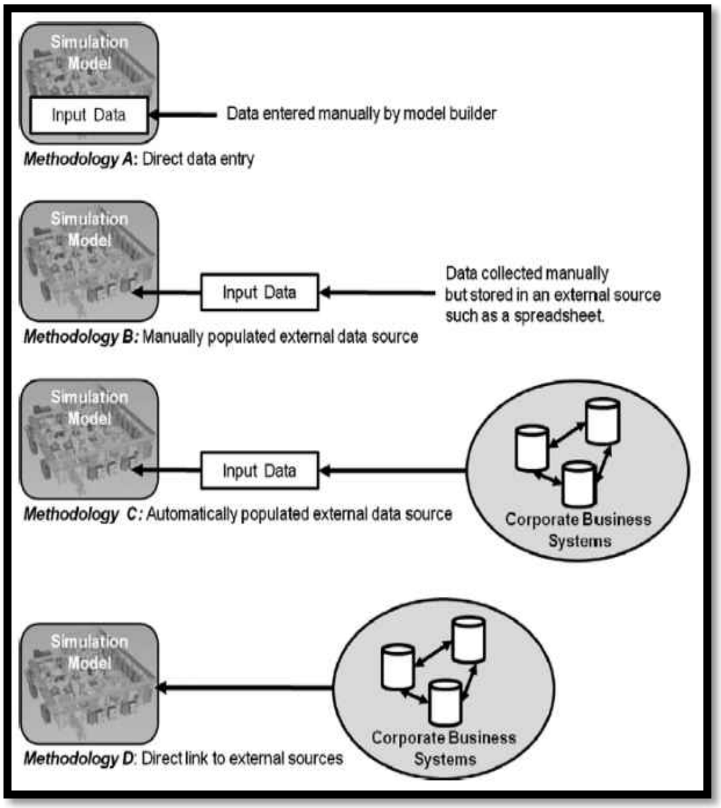

Simulation Model
Input Data
Data entered manually by model builder
Methodology A: Direct data entry
Simulation Model
Input Data
Data collected manually but stored in an external source such as a spreadsheet.
Methodology B: Manually populated external data source
Simulation Model
Input Data
Corporate Business Systems
Methodology C: Automatically populated external data source
Simulation Model
Corporate Business Systems
Methodology D: Direct link to external sources

Figure 1 Alternative data input methodologies.

Source: Skoogh, A. A., Perera, T. T., & Johansson, B. B. (2012). Input data management in simulation - Industrial practices and future trends. *Simulation Modelling Practice and Theory,* 181. doi:10.1016/j.simpat.2012.07.009

In methodology (A) data from several sources are compiled and processed manually; then it is manually entered into the simulation model, so there is not an external data source where the data reside after it is collected. In methodology (B), data and information are stored in an external source, commonly a spreadsheet file, completed manually by the project team, which is then transferred to the simulation [7, 8]. In other words, the DES project team manually performs the collection and processing of data, but unlike methodology (A), these data are not entered directly into the simulation model but maintained in an external data source; then the external data source is linked to the simulation model. Methodology (B) is the most popular approach for input data management, as 61% of surveyed professionals in 2012 use this approach [8].

Methodology (C) uses an intermediate database that feeds the external data source, which feeds the simulation model [7, 8]. The primary difference between methodology (B) and methodology (C) is that the latter involves an automatically-populated external data source from an internal data source (ERP, MPS, etc.), and the former involves a manually-populated external data source. In methodology (D), there are no external data sources because there is a direct link between the internal data sources and the simulation model. Methodologies (C) and (D) solutions can be expensive and complex, especially (D). One reason for this is the fact that there may be a considerable amount of raw data available in the internal systems, but there is limited data that is ready to be used in simulation models [13]. In other words, raw available data often need to be processed further so it can be used in simulation models.

From the same 2012 survey referenced earlier, 80% of those surveyed indicated that their organizations would like to enhance their input data management capabilities so they can implement methodology (C) or methodology (D) within the next 10 years [8]. For instance, the aim for more automation in input data management has motivated research and the development of systems based mostly on methodology (C), such as the Generic Data Management Tool (GDM-Tool), which relies on already available data within internal systems such as ERP. GDM-Tool processes available data so it can be used in simulation projects [14]. Consistent with methodology (C) [8, 14], GDM-Tool not only automates the extraction and preparation of raw data from internal sources, but it also fits statistical probability distributions to the data. GDM-Tool must be properly configured according to the organizational procedures in order to automate data extraction and preparation, which can be challenging and requires trained technical staff [14, 20].

Other documented attempts to automate DES input data management are the Machine Data Acquisition (MDA) and the Parts Data Collection application. MDA for d3FACT simulator, retrieves data automatically and directly from different sources and performs primary processing of the retrieved data following mapping rules, mapping formulas, etc. [15]. The Parts Data Collection application permits the users to pull data from a specific manufacturing database [29], format the retrieved data and feed simulation models. The benefits of the aforementioned solutions focus mainly in manufacturing applications [15, 29], which is one of the several areas where DES is useful. Chapter five presents more details about these documented methodology (C) solutions.

In general, Skoogh et al. present methodology (C) and methodology (D) as “too difficult” [14]. For instance, before any automation, configuration is the first central activity required in setting up the GDM-Tool in order “to map the content of the different data sources to each other and to specify all transformation operations required to obtain the simulation information” [14]. Methodology (C) solutions are more feasible than methodology (D) solutions because the latter requires extensive research, customization and development and are often more complex at the time of implementation [8].

Even though many professionals would like to see a migration towards methodology (C) solutions and to a lesser extent to methodology (D) solutions, the reality indicates that methodologies (A) and (B) are still widely used. As a matter of fact, the most popular approach among DES practitioners is consistent with methodology (B) [8].

Probability Distributions Fitting in DES

After gathering, cleaning and preparing the data to be utilized in DES projects, statistical probability distributions must be fitted to different data inputs in order to perform the simulations using DES software. The fitting of the distributions generates parameters required by the software in order to run one or more replications of the system being analyzed [3]. Nowadays, multi-purpose software such as GDM-Tool [14] can fit most common theoretical statistical probability distributions [17], and there are also specialized, standalone probability distribution fitting software such as Arena's Input Analyzer by Rockwell Automation [4], Stat::Fit® by Geer Mountain Software [18] or ExpertFit by Averill M. Law and Associates [16] that can perform the fitting operation. The time involved when preparing collected data can be an issue with standalone distribution fitting software because these standalone programs expect data files with specific formats or layouts [4]. Obtaining the distributions from a separate standalone software as a separate step is the most common practice [8], especially if the available distributions within the multi-purpose input data management tools are limited [21,17].

In addition to fitting probability distributions to the data, software performing the theoretical distribution fit usually provides a graphical representation of the data in the form of a histogram plus goodness-of-fit tests such as Chi-Square test or Kolmogorov-Smirnov test or Anderson-Darling test [4] in order to confirm the adequacy of different fitted distributions [3,4]. Specialized distribution fitting software also permit empirical distribution fit if no theoretical distribution presents a proper fit based on the goodness-of-fit tests.

Encountering multimodal data is a possibility in DES applications, and, often, this type of analysis has to be performed manually [4]. Current solutions for data fitting display the multimodal data but users need to handle such data manually outside of the probability fitting tool in order to obtain a proper fit on each of the subsets of these data. The solution proposed in this document provides an innovative option to fit multimodal data without having to process each subset manually.

Independence among fitted observations is a key assumption in order to perform statistical goodness-of-fit tests and to use maximum likelihood estimation (MLE) to obtain probability distribution parameters [3]. Currently, not all statistical probability distribution fitting software offer the capability to test, diagnose or visualize autocorrelation [4,16,18]. The solution proposed in this document offers the ability to test, diagnose and visualize autocorrelation in real-time right after new data are collected.

Analysis of Input Data Management Methodologies

The different approaches or methodologies described above explain that solutions based on methodology (C) and on methodology (D) require data sources already in place [14, 15]. In other words, such solutions are not meant to be used in the collection of the data but assume availability of accessible data. Such reliance on existing data poses important limitations to the proliferation of these types of solutions. On the other side of the spectrum, manual approaches such as methodology (A) and methodology (B) stress manual data collection and manual data preparation in order to feed simulation software, without much attention paid to automation that facilitates their implementation.

Methodology (C) solutions add automation to methodology (B) in the form of data updates coming from an internal data source. Even though methodology (C) implementations are a desirable goal of 80% of surveyed professionals in 2012 [7, 8], literature shows few of such developments. In addition to the reliance on available data, another reason why sophisticated solutions such as (C) and (D) are not widely implemented is the fact that they are “too difficult”, according to Skoogh, Perera and Johansson [8]. Methodologies (C) and (D) solutions require detailed customization and complex implementation rules in order to convert internal data into simulation ready output.

The fact that methodology (A) and methodology (B) are the most common approaches to collect data suggests that they present features that are convenient to DES modelers. Equivalently, it can be deducted that methodology (C) and methodology (D) solutions present complexities that prevent them from being implemented on a consistent basis.

One of the strengths of performing manual collection of data is that the collector has control over the inputs and these can be verified at any time [8]. Methodology (A) and methodology (B) solutions tend to be fairly general and work in several applications of DES. A common disadvantage of manual approaches, such as methodology (A) and methodology (B), is that they can be time-consuming [2]. In addition, manual collection of data for DES models can carry human error and become a tedious activity when entering values and processing the data.

Potential Hybrid Solution for Input Data Management

After analyzing the strengths and weaknesses of documented methodologies it is evident that there is room for exploring combinations of features from the different methodologies in order to find an alternative solution to the ones documented in the literature. **Figure 2** presents the idea of a hybrid solution for DES input data management.

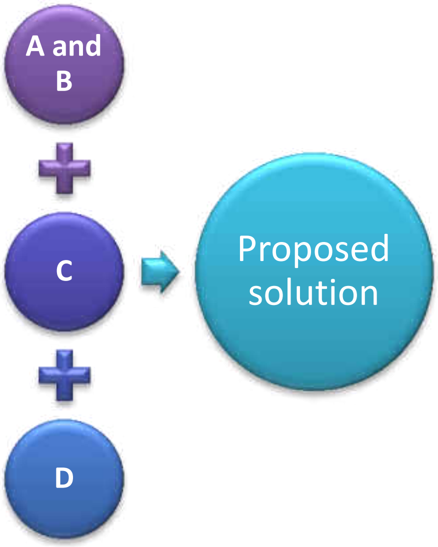


Figure 2 Potential hybrid solution combining features of different methodologies

For instance, a desirable feature of a potential new input data management solution is for it to be general enough so it can be used in different types of DES projects, in different disciplines, beyond manufacturing. Another desirable feature of a potential new solution is that it should be easy to configure and implement. Adding to the list of desirable features, it would be convenient to have a solution that allows the collector to verify the data at any time.

The desirable features just described are consistent with the advantages of methodologies (A) and (B), the latter being the most popular of all input data management approaches [8]. As mentioned earlier, the main disadvantage of methodologies (A) and (B) is the manual collection and preparation of data, hence a new solution should avoid tedious, error prone manual processing of data. In other words, any potential new solution should aim for a way to automate the collection, validation, storage, processing, visualization and fitting of input data with a simple action such clicking with a computer mouse.

As stated previously, even if data are already available internally, these data are often not ready for DES projects. Hence, even though a solution starting at the collection stage can be regarded as more time consuming than a solution getting the data directly from internal systems, the fact that previously available data often need considerable processing after extraction [8,21] suggests that the time-consumption benefit of working with already available data might not be as significant as expected. In general, available data accessed by methodology (C) and methodology (D) solutions are the result of complex implementation and configuration [8], which represents an obstacle to the proliferation of these types of solutions [22]. So even though internal systems produce large amounts of raw data, the fact that these data have not been designed or prepared for DES projects [8,21] can elevate costs relative to working with data collected outside of the internal systems and specifically designed for DES projects.

Enhancements beyond Methodology (B)

In order to make a new solution based on the strengths of methodologies (A) and (B) more attractive than relatively sophisticated methodology (C) and methodology (D) solutions, it should feature some of the key conveniences found in methodology (C) and to a lesser extent in methodology (D) without losing the conveniences of methodologies (A) and (B). One of these convenient features is the ability to update existing databases automatically without the need of human assistance [8] as new data are generated. In other words, a new solution should provide a method to replace internal data systems from (C) and (D) with a mechanism of data generation that automatically feeds the external data sources, which feed the simulation models.

Another useful feature that can be added to the new input data management solution is the ability to create histograms and fit probability distribution that update in real-time after a new observation is recorded. The updated histogram and probability distribution fit should be visible to the data collector right after the new data are added. The probability distribution fit should present the usual goodness-of-fit tests [3]. Moreover, the new solution for input data management should feature a visual way of notifying the analyst about the goodness-of-fit without having to refer to the goodness-of-fit test results every time the data are updated. In addition to the desirable features described above, a potential new solution should also offer the ability to test, diagnose and visualize serial correlation among observations in real-time in order to validate underlying theoretical assumptions [3].

In general, the new solution should be intuitive and consistent with the principles of human-computer interaction design [22, 34, 35]. A new solution should be flexible enough to accommodate different characteristics of the data being collected regardless of the discipline or area of application.

Similar to methodology (C) solutions, the new potential input data management solution should automatically record, update and store the data in a separate location, such as a database or a data file. Similar to methodology (D) solutions, the hybrid solution should produce output that is definite and meaningful in the context of DES projects. In other words, the hybrid solution should produce output that is ready to be added to DES software via automated imports or by entering the resulting parameters into the software [14, 21, 23- 28]. Automated imports into DES software is outside the scope of this research work.

In summary, there is the need for a new hybrid input data management solution that combines the specific strengths of commonly-used manual methodologies (A) and (B) with features of the more complex methodologies (C) and (D). This research investigation seeks to provide the theoretical foundations for developing the aforementioned hybrid solution in order to solve the MDCAP problem prevalent in DES projects.

CHAPTER THREE: METHODOLOGY

Formulation of the MDCAP Problem

The manual data collection and processing (MDCAP) problem in the context of DES projects can be formulated as follows.

On average 31% of DES projects' time is devoted to data collection and processing. In some instances this percent can go up to 40%. The main reason for this is the fact that input data management for DES projects is a highly manual process [7]. MDCAP includes steps such as collecting the data, updating existing data, performing calculations on the data, validating the data, copying and pasting the data in different formats, visualizing the data, fitting and testing probability distributions to the data, storing the data and calculating relative frequencies of distinct outcomes at different decision points within the system being modeled.

As mentioned earlier in this document, 80% of professionals or practitioners rely on manual methodologies (A) and (B), which shows that manual approaches to data collection and processing are the most common practice in the field. If this practice persists, the time devoted to input data management within DES projects will likely stay at roughly one third of the project's lifecycle. Given the fact that more sophisticated solutions such as the ones from methodologies (C) and (D) are considered "too difficult" [8] also suggests that the reliance on manual approaches for data collection and processing will not go away. The fact that 80% of surveyed professionals also affirmed that they would like to migrate towards automated solutions for input data management suggests that methodologies (A) and (B) are not the preferred practice if these can be avoided, even though methodologies (A) and (B) present certain advantages such as verifiability and transparency.

The MDCAP problem is the negative impact on time-to-completion and on user experience caused by manual approaches to input data management in DES modeling projects. The MDCAP problem is generated by the inefficiencies associated with manually collecting and processing the data necessary for DES projects.

Proposed Framework for Automated Data Collection and Processing

Chapter one generalized DES input data into three specific data classes given the model logic. Such generalization constitutes the foundation for the theoretical framework detailed in this chapter and is an original contribution of this research investigation to the body of knowledge. Classifying DES input data as Class (I), Class (II) or Class (III) permits the intended automation and the applicability of the proposed framework in different disciplines where DES modeling is appropriate.

Theoretical Foundations

The idea of a hybrid solution to automate DES input data management is presented **Figure 2**. **Figure 2** suggests combining the strengths of different methodologies in order to generate a user-friendly, intuitive solution that automates input data management for DES projects.

The three generic data classes feed DES models based on a predefined model logic. In what follows, model logic is assumed to be known before starting the collection of the data. The three proposed DES input data classes constitute the key foundation in the development of an automated alternative framework to address the MDCAP problem. Formally, these classes are defined as follows.

Class (I) data: data where

$$t_{end,j} = t_{start,j+1} \tag{3.1}$$

For consecutive events indexed *j* and *j*+1, the ending timestamp associated with event *j*, $t_{end,j}$ represents the start timestamp of the next event (*j*+1), $t_{start,j+1}$.

Class (II) data: data where

$$t_{end,j} \neq t_{start,j+1} \tag{3.2}$$

For consecutive events indexed *j* and *j*+1, event *j* has a starting timestamp and an ending timestamp for a given event, represented by $t_{start,j}$ and $t_{end,j}$, respectively; For consecutive events *j* and *j*+1, the ending timestamp of event *j*, $t_{end,j}$, does not represent the starting timestamp of the next event *j*+1, $t_{start,j+1}$.

Class (III) data: data where tallied counts generate relative frequencies or probabilities

$$\gamma_{im}=\frac{n_{im}}{N_m} \tag{3.3}$$

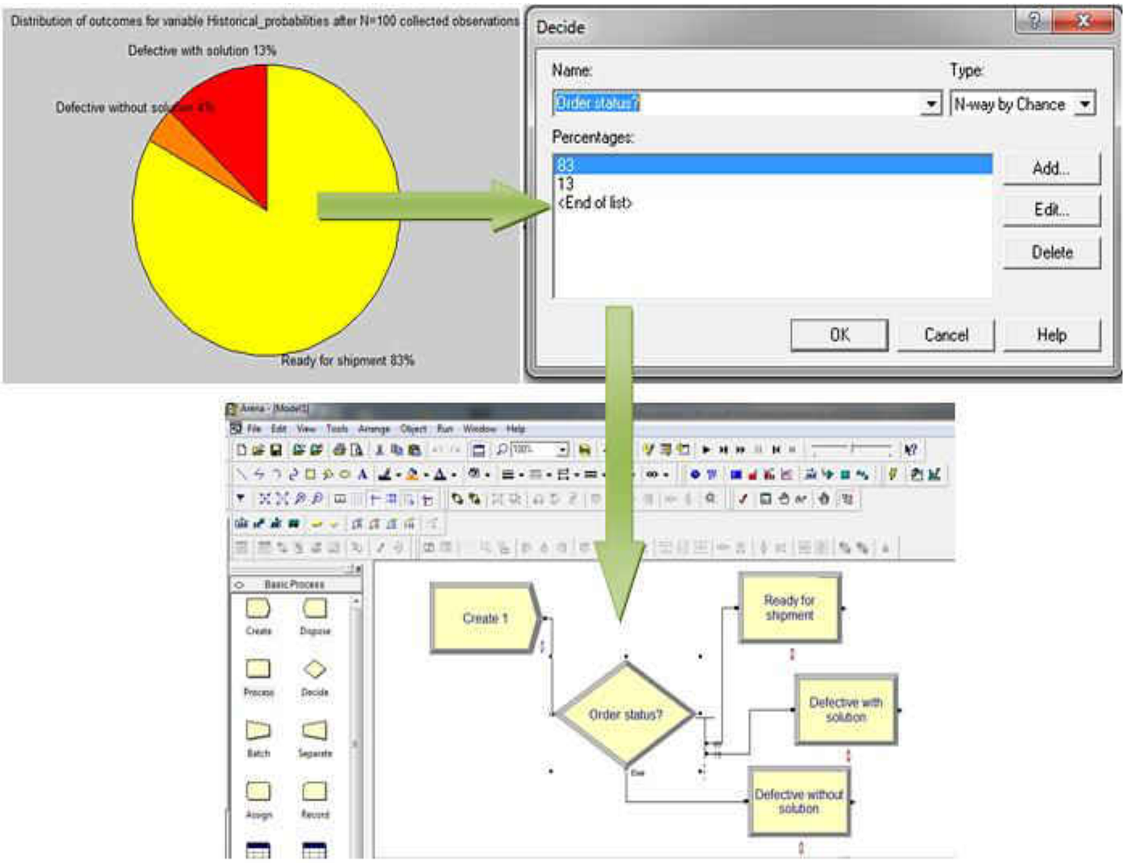
...where $m=1,...,M$ and $i=1,...,c_m \; \forall \; m$

$$\sum_{i=1}^{c_m} \gamma_{im} = 1 \tag{3.4}$$

...at each decision point m, $m=1,...,M$

γ_{im} represents the probability of the specific outcome *i* out of c_m distinct possible outcomes, at the specific decision point *m,* so $i=1,...,c_m$ and $m = 1,\ldots, M$. Total counts for each of the c_m distinct outcomes at decision point *m* are represented by n_{im} , $i = 1, \ldots, c_m$. Class (III) data are counts n_{im} for each of the c_m distinct outcomes out of the N_m total observations tallied at the decision point *m*. The counts n_{im} determine relative frequencies for each of the c_m distinct outcomes at decision point *m* as a proportion of N_m. From these elements, relative frequencies for each distinct possible outcome at a given decision point *m* can be computed as shown in (3.3). **Figure 3** shows an example of how the γ_{im} ‘s become inputs in Arena DES software.

In the example from **Figure 3**, $c_m = 3$, as there are three possible outcomes, e.g. defective with solution, defective without solution and ready for shipment. There is only one decision point, hence m=1 and M=1. Also, $\gamma_{11} = 0.83$, $\gamma_{21} = 0.13$ and $\gamma_{31} = 0.04$. The way to interpret γ_{11} is that the probability of the first distinct outcome at decision point 1 is 0.83. The way to interpret γ_{21} is that the probability of the second distinct outcome at decision point 1 is 0.13. The way to read γ_{31} is that the probability of the third distinct outcome at decision point 1 is 0.04.


Distribution of outcomes for variable Historical_probabilities after N=100 collected observations
Defective with solution 13%
Defective without solution 4%
Ready for shipment 83%
Decide
Add...
Edit...
Delete
OK
Cancel
Help
Arena - [Model1]
File Edit View Tools Arrange Object Run Window Help
Basic Process
Create
Dispose
Process
Decide
Batch
Separate
Assign
Record
Create 1
Order status?
Else
Ready for shipment
Defective with solution
Defective without solution

Name: Order status?

Type: N-way by Chance

Percentages:
83
13
<End of list>

Figure 3 Example of how historical probabilities are used in the Arena DES software.

Claim 3 Let Ω be the set of disciplines where DES is applicable. Let δ_ρ be a data collection approach to gather input data for DES project ρ, and let Φ be the model logic of the system being analyzed using DES within the project ρ, then

$$\delta_\rho\,(\Phi|\,\Omega) = \delta_\rho\,(\Phi) \tag{3.5}$$

This means that the data collection approach for DES project ρ is a function of the model logic Φ and is independent of the area of application or discipline represented by Ω.

Independence between Φ and Ω

Because the collection of data for similar model logic is the same, regardless of the discipline, a single data collection tool that covers Class (I), Class (II) and Class (III) can be used to collect data in any discipline from Ω, given the model logic Φ. This can be summarized as:

Claim 4 If a data collection approach δ_ρ can collect Class (I), Class (II) and Class (III) data, this approach can be used in any discipline ω that belongs to Ω, in order to collect input data for a DES project ρ, given model logic Φ. Then,

$$\rho = f\,(\delta_\rho|\Phi) \tag{3.6}$$

Hence, if there is a model logic Φ in place and data from Class (I) and/or Class (II) and/or Class (III) are collected, a DES project ρ will be feasible, regardless of the discipline or area of application. Another interpretation of (3.6) is that specifying the discipline where a DES project ρ will be performed is not enough to create the DES project ρ; whereas if there exists a model logic Φ and data gathered using approach δ_ρ, the DES project ρ will be feasible.

Resuming a Previous Data Collection Effort

Resuming a previous data collection effort without distorting the information from previously gathered data is a key requirement in order to achieve automation in DES data collection and processing. There needs to exist the capability to add new data to previously collected data without corrupting the information obtained from older data. As will be presented below, Class (I) data are susceptible to the timing of the collection efforts, so there needs to be a mechanism to neutralize the impact from resuming data collection at a later point in time.

For example, imagine that an analyst is recording arrival times to an airport security checkpoint in order to compute interarrival times (e.g. Class (I) data). The analyst takes lunch break and resumes collecting arrival times after finishing lunch. When he or she comes back from lunch, the specific interarrival time calculated as the difference between the latest arrival timestamp recorded before lunch and the first arrival timestamp recorded after lunch will be a corrupted interarrival time. The data corruption just mentioned will have a direct impact in the probability distributions and eventually in the simulations generated from these data.

Let us define a data collection effort as the uninterrupted action of collecting data to be used in DES projects. This means that if the data collector quits collecting data for any reason and then resumes this collection at a later point in time, then the collector has engaged in two collection efforts. There is no limit in the number of efforts for a given DES project.

Let $\varepsilon_{i\psi\vartheta} > 0$ be i^{th} effort to collect class ψ data for variable ϑ for a DES project, where ψ can take the values *I*, *II* or *III* depending on whether the collection is on Class (I), Class(II) or Class(III) data. Let $\tau_{i\psi\vartheta}$ be a timestamp from $\varepsilon_{i\psi\vartheta}$, in order to gather class ψ data for variable ϑ. The rules to resume data collection are presented below.

Let Δ_{ψ} be defined as the act of resuming data collection for class ψ data, where ψ can take the values *I*, *II* or *III*.

If $\psi = III$ then resuming the data collection does not depend on the effort of the collection or in the timestamp of the collection, only on model logic. Formally

$$\Delta_{III}(\Phi) \tag{3.7}$$

If $\psi = II$ then resuming the data collection does not depend on the effort of the collection, only in the timestamps of the collection and on model logic. Formally

$$\Delta_{II}(\Phi, \tau_{iII\vartheta}) \tag{3.8}$$

If $\psi = I$ then resuming the data collection depends on the effort of the collection, on the timestamp of the collection and in the model logic. Formally

$$\Delta_{I}(\Phi, \tau_{iI\vartheta}, \varepsilon_{iI\vartheta}) \tag{3.9}$$

For Class (II) data, each collection is generated by two timestamps, referred here as starting timestamp and ending timestamp. After a collection is completed from recording a starting timestamp and an ending timestamp, then the data collection can be resumed anytime in the future without affecting previous distribution of Class (II) data. Hence, resuming the collection of Class (II) data does not depend on the effort $\varepsilon_{iII\vartheta}$.

For Class (I) data each entry consists of a single timestamp, where new timestamps are compared to the previously recorded timestamp in order to generate an interarrival collection. If the last recorded timestamp was gathered on a different effort (as in the simple airport security example presented earlier), then the time between collection efforts will corrupt the recording of Class (I) data. Hence, the resuming of Class (I) data needs to be independent of collection efforts $\varepsilon_{iI\vartheta}$'s. In other words, the following condition must hold:

$$\Delta_{I}(\Phi, \tau_{iI\vartheta} | \varepsilon_{iI\vartheta}) = \Delta_{I}(\Phi, \tau_{iI\vartheta}) \tag{3.10}$$

In order to achieve condition (3.10) from (3.9) the following logic needs to be applied when resuming Class (I) data collection and processing.

Let us introduce two consecutive efforts to collect Class (I) data for variable ϑ, namely $\varepsilon_{iI\vartheta}$ and $\varepsilon_{(i+1)I\vartheta}$, where I in the subscript stands for Class (I) data and i represents the effort. Let E be the set of all the collection efforts before i+1, so $\varepsilon_{iI\vartheta} \in E$. Let the latest timestamp value from the collection effort $\varepsilon_{iI\vartheta}$ be called $t_{N,\text{original}}$. When the next effort starts, namely $\varepsilon_{(i+1)I\vartheta}$, $t_{N,\text{original}}$ gets converted into the exact timestamp when $\varepsilon_{(i+1)I\vartheta}$ starts. Once converted, $t_{N,\text{original}}$ is renamed to $t_{1,\text{resumed}}$. Then the positive difference between $t_{N,\text{original}}$ and $t_{1,\text{resumed}}$ is calculated. This positive difference is represented by $\sigma_{(i+1)\vartheta}$ for each resuming effort $\varepsilon_{(i+1)I\vartheta}$, Formally,

$$\sigma_{(i+1)\vartheta} = t_{1,\text{resumed}} - t_{N,\text{original}} \tag{3.11}$$

… for the $(i+1)^{\text{th}}$ collection effort $\varepsilon_{(i+1)I\vartheta}$ of Class (I) variable ϑ

$$\forall\ \varepsilon_{iI\vartheta} \in \mathrm{E}$$

Time difference $\sigma_{(i+1)\vartheta}$ needs to be added to all the previously recorded timestamps collected before $t_{N,\text{original}}$. The new timestamps obtained after adding $\sigma_{(i+1)\vartheta}$ are used to calculate the new set of interarrival times. All the previous arrival timestamps recorded before $\varepsilon_{(i+1)I\vartheta}$ get modified by $\sigma_{(i+1)\vartheta}$, but the Class (I) interarrival times remain the same regardless of the collection effort, fulfilling (3.10).When the second entity appears in the resumed collection effort $\varepsilon_{(i+1)I\vartheta}$, at $t_{2,\text{resumed}}$, the difference between consecutive timestamps $t_{2,\text{resumed}}$ and $t_{1,\text{resumed}}$, becomes the first interarrival time of the resumed collection effort $\varepsilon_{(i+1)I\vartheta}$. Then Class (I) data collection can continue without any corrupted data.

Best Fitted Statistical Probability Distribution

Let π be the non-empty set of probability distributions fitted to the resulting histogram of N observations from Class (I) or Class (II) data. Let *K* be the subset of π where the null hypothesis of good fit is not rejected. Then the best fitted probability distribution out of *K* is obtained as follows [34].

- If $N < 50$ then λ is the fit that produced Max (Max (p-value $_{KS}$), Max (p-value $_{AD}$)) $\forall\ K \epsilon \pi$
- If $N \geq 50$, then λ is the fit that produced Max ($\text{p} - \text{value}_{\chi^2}$) $\forall\ K \epsilon \pi$

λ represents the best fitted probability distribution out of $K \in \pi$, *K* is the set of distributions from π where the null hypothesis of good fit is not rejected, *AD* stands for Anderson-Darling goodness-of-fit test, *KS* stands for Kolmogorov-Smirnov goodness-of-fit test and χ^2 stands for the Chi Square goodness-of-fit test.

Speed between Two Points Using Class (II) Data

A common data input to DES projects is the speed of different actors within the simulation. Entities, resources, transporters and other simulation elements often need to be assigned a speed when they move from one location to another within the modeled system. The proposed framework permits the automated calculation of average speed between two points using the reasoning behind the collection and processing of Class (II) data. Because Class (II) data can be regarded as having a departure timestamp and an arrival timestamp instead of starting and ending timestamps, the average speed s between two points can be deducted by knowing the traveled distance D and using formula (3.12). Let t_i , i=1,…,d be one of the entries of $d\ x1$ vector ς of calculated Class (II) continuous time intervals. Then, a $d\ x1$ vector of speeds S, with elements s_i, i=1,…,d, can be obtained using

$$s_i = u \bullet \frac{D}{t_i} \tag{3.12}$$

D is the target distance, t_i represents the i^{th} Class (II) time interval within vector ς and u is a conversion constant. If speed is in miles per hour, D in meters and time in minutes, u= 0.03728.

Expected Impact in DES Input Data Management

This section presents the expected gains from using the proposed framework instead of using MDCAP associated with dominant input data methodologies (A) and (B) and instead of using more sophisticated solutions based on methodologies (C) and (D).

A critical concept for this analysis is “the unavoidable wait for events to happen”. The unavoidable wait for events to happen can be defined as the unavoidable human wait associated with data collection for DES projects. For instance, if an analyst is collecting data on car arrivals to a gas station, there is an unavoidable wait for cars to arrive. The proposed framework reduces the time associated with DES input data management to be equal to the unavoidable wait for events to happen, eliminating all other sources of delays associated with data processing after the data are collected.

Let β_ρ >0 be the total time devoted to input data management for DES project ρ. Let $\eta_\rho \geq 0$ be the unavoidable wait for events to happen for DES project ρ. Let $\theta_\rho \geq 0$ be the time necessary for implementing an input data management solution for DES project ρ. Let $\Lambda_\rho \geq 0$ be the time devoted to post-collection data processing for DES project ρ.

Under MDCAP β_ρ can be broken down in two components because the configuration component $\theta_\rho \approx 0$. Under MDCAP θ_ρ is considered to be almost equal (≈) to 0 because usually MDCAP is performed by standalone specialized tools that are ready to be used after simple installation steps. The first component of β_ρ under MDCAP is the unavoidable wait for events to happen $\eta_\rho > 0$ and the second component includes a series of processing steps beyond data collection represented by $\Lambda_\rho > 0$. The post-collection component Λ_ρ includes, but is not limited to, data calculations, data reformatting, data exports, data imports, validations and other processing steps often performed by standalone isolated tools. Hence, under MDCAP,

$$\beta_\rho \approx \eta_\rho + \Lambda_\rho \quad (3.13)$$

$$\forall \rho$$

Under sophisticated methodologies (C) and (D), the unavoidable wait for events to happen and the post-collection processing time can be eliminated due to the automated extraction and processing of available internal data. Hence, if there exists internal data, under sophisticated (C) and (D) solutions $\Lambda_\rho \approx 0$ and $\eta_\rho \approx 0$. In these types of sophisticated solutions for automated DES input data management there is a significant amount of time that is required for implementation and configuration in order to produce simulation-ready data [8]. This implies that for methodology (C) and (D) solutions $\theta_\rho > 0$. Hence, for methodology (C) and (D) solutions, if there exists available data within internal data systems,

$$\beta_\rho \approx \theta_\rho \quad (3.14)$$

$$\forall \rho$$

Claim 5 Under the proposed integrated framework for DES data collection and processing all delays in project ρ other than η_ρ are eliminated. Formally, under the proposed framework

$$\Lambda_\rho \approx 0 \tag{3.15}$$

$$\theta_\rho \approx 0 \tag{3.16}$$

Hence, given model logic, under the proposed framework

$$\beta_\rho \approx \eta_\rho \tag{3.17}$$

$$\forall \rho$$

Claim 6 For a given DES project ρ and focusing only on time-to-completion, the proposed framework will be preferred to MDCAP if $\Lambda_\rho > 0$; the proposed framework will be preferred over sophisticated methodology (C) and (D) solutions if $\theta_\rho > \eta_\rho$.

Opportunity for Innovation

The proposed framework not only permits the automated collection and processing of Class (I), Class (II) and Class (III) data, but also facilitates the implementation of other useful features needed in DES analysis such as autocorrelation diagnostics, multimodal data visualization and transporter speed computations.

The complex nature of methodology (C) and (D) solutions opens the window for a new automated solution based on the proposed integrated framework for automated DES data collection and processing. The automated solution developed following the proposed framework intends to streamline and integrate DES data collection, data updates, data storage, data processing, data visualization, data validation, distribution fitting and distribution testing in real-time, in response to intuitive user actions. Such streamlined integration is what makes a solution based in the proposed framework unique in the field of modeling and simulation.

Human-Computer Interaction Design Considerations

This research investigation includes the programming of a user interface based on the concepts and ideas of the proposed integrated framework. Hence human computer interaction (HCI) design principles must be considered when programming the aforementioned interface. As previously mentioned, the intended interface can be regarded as a hybrid that includes features from different input data management methodologies (A), (B), (C) and (D). The main user experience goals of this automated solution to MDCAP are

- Avoidance of the undesirable characteristics of MDCAP;
- Integration of currently isolated data processing steps into a single information stream triggered by a simple user action;
- Streamlined validation of the results when new data are generated; and
- Reliable data storage.

The design of the proposed interface to automate DES data collection and processing should increase productivity and user satisfaction [35]. The proposed interface must be "usable". Being "usable" in HCI design means that the proposed solution to MDCAP must be easy to learn and effective to use, while providing an enjoyable user experience [22]. Any innovative interface that seeks to improve DES data collection, data processing, data storage, data updates, validation of theoretical assumptions and distribution fitting must reliably provide correctness, no duplication, consistency, timeliness, validity, reliability and completeness [7, 29, 30, 31, 32].

The interface that instantiates the proposed integrated framework for DES data collection and processing should take into account the following principles from HCI design [22]:

- Visibility: functionalities are clearly defined and easy to identify.

- Feedback: the user receives a specific response to different types of user inputs.
- Constraints: restrict user interaction at a given moment.
- Consistency: no ambiguity in the expected response to user inputs.
- Affordance: hints the user on how to use the interface even without instructions.

HCI design also indicates that interfaces should be satisfying, enjoyable, engaging, exciting, helpful, motivating, aesthetically pleasing, supportive of creativity, cognitively stimulating, rewarding and emotionally fulfilling.

The conceptual model [22] for the proposed interface is the following: a system that offers the ability to declare a variable that will be used in a DES model; the declared variable then gets collected, stored, processed and updated in real-time, right after pressing specific buttons within the interface using a computer mouse.

The interaction type with the proposed interface needs to be "instructing", where users issue instructions to the system [22]. The instructions should be given by first specifying the name of the variable being collected, second, by checking boxes to determine optional parameters and third, by pressing buttons to initiate the real-time collection, storage, processing and updating of the data. User inputs should generate a response from the system that is relevant to DES models, including images of the expected information.

The declaration of a variable should be by typing the variable's name into a space within the interface. The reason for typing the variable's name into a space is to allow users to define a name that is appropriate for each DES project. If variables' names were entered as selections from a dropdown or by checking a box, variables' names would be restricted to a small set of options, whereas if the names are typed, the options are limitless to the user. Instructions provided via box-checking allow the user to intuitively select one or more options from a relatively small, fixed set of options [22]. Variable declaration and box checking precede button pressing.

The reason for using button pressing to initiate the automated processing (and not other options like drop downs, box checking or radial buttons) is because buttons are an intuitive way to issue a definite command. Buttons yield a sense of definiteness and closure (i.e. GO, SUBMIT, NEXT, etc.) [22].

In order to make this interface easy and pleasant to use, the interface should reduce visual work, reduce intellectual work, reduce memory work, reduce motor work and eliminate or minimize burdens imposed by technology [35]. Other considerations in the design of the proposed user interface to address MDCAP include high color contrast, distinctive colors used with specific cues (i.e. brief instructions) and opponent colors separated from each other [36].

Instantaneous feedback with relevant information for DES models should be provided in the form of message boxes [22] and relevant graphics. Messages such as errors or warning should be clearly presented as a separate box, in the center of the screen and not in the peripherals, so they can be easily visualized [36]. For instance, errors can be presented in red, with an “X” and warnings in yellow, with a “!” [36].

Real-time responsiveness should be a key characteristic of this design [36] because users must receive feedback as soon as they collect new data. In other words, processing results should display real-time confirmations of the data collection and generate real-time output relevant to DES models. Relevant output for DES models includes probability distribution fitting, histograms, autocorrelation diagnostics, special options for visualizing multimodal data and pie charts, among others. A user interface instantiating the proposed framework is presented in chapter four and tested in chapter six.

CHAPTER FOUR: RESULTS

Overview of the Prototype Interface

Results are focused on the ability to generate an automated solution for input data management based on the proposed framework. The resulting prototype application is called DESI, which stands for Discrete Event Simulation Inputs. DESI is a graphical user interface (GUI) that is used to collect and process data for DES projects. The data to be collected are determined by the model logic. The proposed user interface has three main sections and a section of optional parameters. The three main sections of DESI are the top blue section, used to collect, fit and test Class (I) data; the middle orange section, used to collect, fit and test Class (II) data; and the bottom green section, used to collect and display Class (III) data. The flow of information when using DESI is detailed in **Figure 4**, starting from the top.

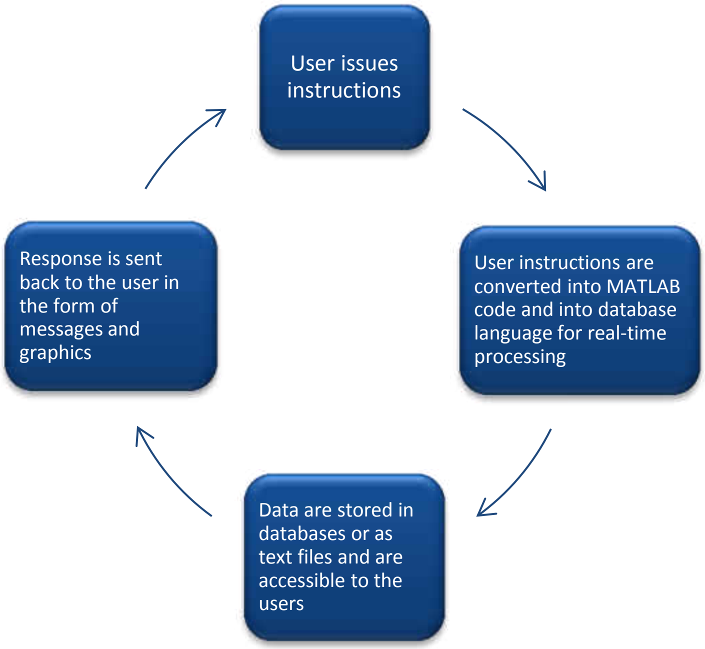

User issues instructions
User instructions are converted into MATLAB code and into database language for real-time processing
Data are stored in databases or as text files and are accessible to the users
Response is sent back to the user in the form of messages and graphics

Figure 4 The flow of information when using the DESI interface.

There are two versions of DESI, DESI 1.0 and DESI 2.0. DESI 1.0 relies on database connectivity in order to store the data in MySQL tables. DESI 2.0 stores the data in the form of text files and presents key enhancements for DES data preparation beyond DESI 1.0. DESI 1.0 has safer data storage, whereas DESI 2.0 presents more features, a simpler implementation and can be installed on both mac and pcs. DESI 2.0 evolved from DESI 1.0.

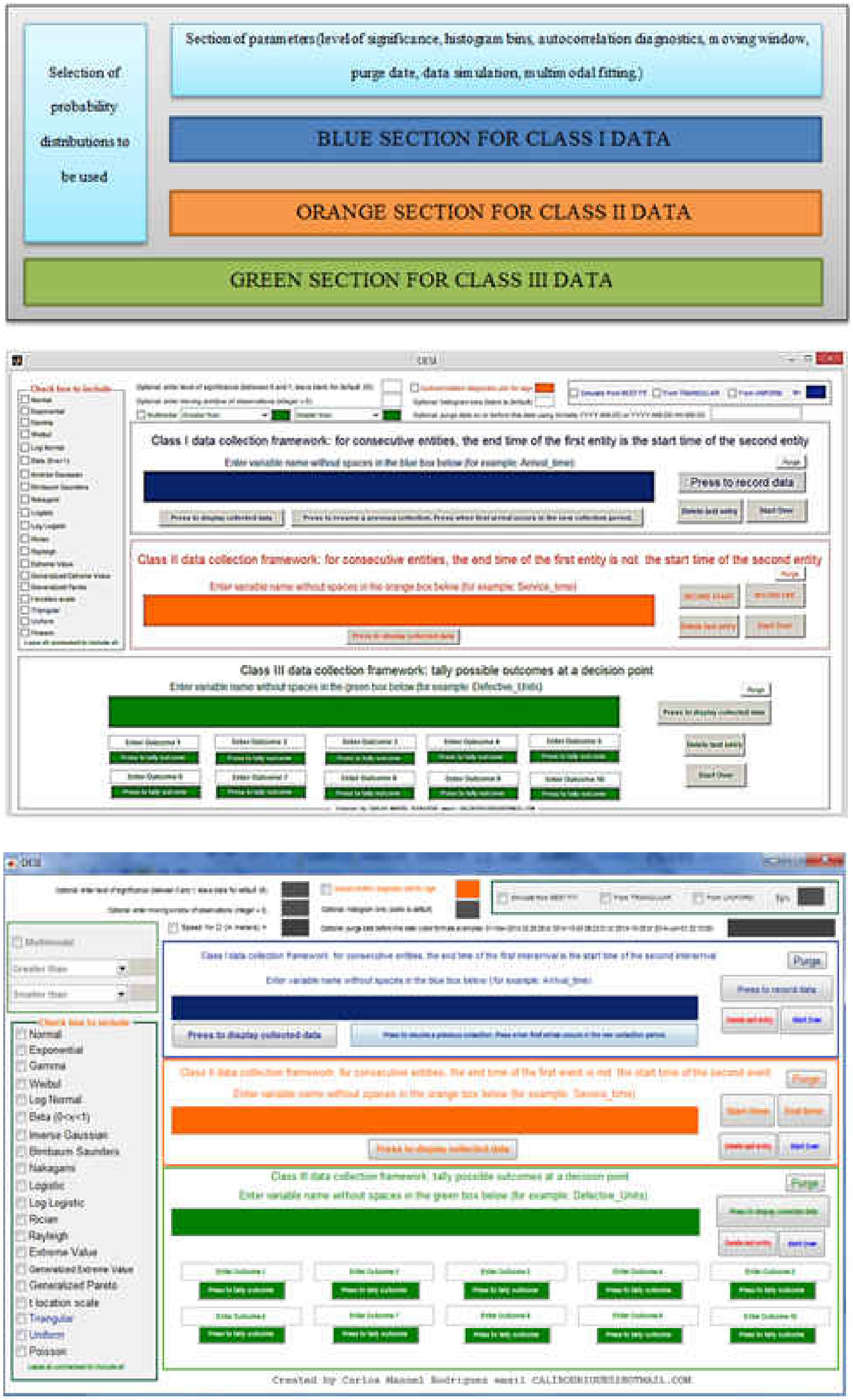
The DESI interface sketch, the DESI 1.0 graphical user interface and the DESI 2.0 graphical user interface are shown in **Figure 5.** The user issues instructions to both versions of DESI in the form of

a) Box checking (to select what probability distributions to fit, to process subsets of the data, to display autocorrelation diagnostic plots, to calculate transporter speed[1] and to simulate new data based on different types of fit);

b) Typing (to declare variables, to determine moving window of most recent data, to fix the number of histogram bins, to determine ranges of data to process, to specify number of lags for autocorrelation diagnostic plots, to set a cutoff date for data purging, to enter distances for speed calculations[1] and to provide the number of simulated data points to be generated from different fits); and

c) Button pressing (to start procedures).

[1] Feature only available in DESI 2.0.


Selection of probability distributions to be used
Section of parameters (level of significance, histogram bins, autocorrelation diagnostics, moving window, purge date, data simulation, multimodal fitting)
BLUE SECTION FOR CLASS I DATA
ORANGE SECTION FOR CLASS II DATA
GREEN SECTION FOR CLASS III DATA
DESI
Check box to include
Press to record data
Purge
Start Over
Delete last entry
Press to display collected data
Enter Outcome 1
Press to tally outcome
Multimodal
Greater than
Smaller than
Normal
Exponential
Gamma
Weibul
Log Normal
Beta (0<x<1)
Inverse Gaussian
Birnbaum Saunders
Nakagami
Logistic
Log Logistic
Rician
Rayleigh
Extreme Value
Generalized Extreme Value
Generalized Pareto
t location scale
Triangular
Uniform
Poisson

Class I data collection framework: for consecutive entities, the end time of the first entity is the start time of the second entity

Enter variable name without spaces in the blue box below (for example: Arrival_time)

Class II data collection framework: for consecutive entities, the end time of the first entity is not the start time of the second entity

Enter variable name without spaces in the orange box below (for example: Service_time)

Class III data collection framework: tally possible outcomes at a decision point

Enter variable name without spaces in the green box below (for example: Defective_Units)

Created by Carlos Manuel Rodriguez email CALIRODRIGUEZ@HOTMAIL.COM

Figure 5 DESI interface sketch, DESI 1.0 GUI and DESI 2.0 GUI.

Both versions of DESI offer the ability to display previously-recorded data, delete the most recent observation in case it was erroneously recorded, resume a previous collection, test serial correlation, analyze multimodal data, select what probability distributions to fit and simulate new data based on different fits. DESI can display the most recent observations as a moving window in case the analyst is only interested in the most recent activity. Both versions of the proposed tool also have the option to purge data before a specified cutoff date in order to avoid storing old, unnecessary data.

Because the count of the collected data values will not be large in comparison to the storage and processing capabilities of databases, installing the databases on personal computers suffices [38, 39, 40, 41] in order to run DESI 1.0. DESI 2.0 stores the data as text files so system memory should not present any constraint.

Probability Distribution Fitting and Determination of Best Fit

As presented in chapter three, the Kolmogorov-Smirnov and Anderson-Darling goodness-of-fit tests are performed if the number of fitted observations is less than 50 and the Chi Square goodness-of-fit test is performed if the number of observations is greater than or equal to 50 [34].

In both versions of DESI, the analyst determines the set π of probability distributions to be fitted to the resulting histogram from Class (I) or Class (II) data. The user determines π by checking boxes in the left panel of DESI or if all boxes are left unchecked, all distributions will be included. In this project, π can have some or all of the following distributions: Normal, Lognormal, Gamma, Exponential, Weibull, Triangular, Uniform, Beta, Birbaum-Saunders, Extreme Value, Generalized Extreme Value, Generalized Pareto, Logistic, Log Logistic, Nakagami, Rayleigh, Rician, T Location Scale, Inverse Gaussian (also known as Wald) and Poisson. Ideally, the user is expected to select the probability distributions supported by the DES software being utilized to complete the DES models. Distribution parameters are obtained using Maximum Likelihood Estimation (MLE) [37], which assumes independence between observations [3].

The information produced by both versions of DESI can be added to DES software or it can be modified further by collecting more data values. In what follows, if DESI is referenced without the version number, then the statements are true for both versions of DESI.

Development of DESI 1.0

The ideas and concepts of the proposed integrated framework were initially implemented using a combination of Matlab code and MySQL databases in order to create the DESI 1.0 interface. The reason for using Matlab is its ability to handle vectors, probability distributions, goodness-of-fit tests and graphics. In addition Matlab can connect to databases using the database toolbox capabilities [37]. The reason for using MySQL is because it is a well-documented open source tool, widely used in different industries [38- 41]. There are other programming languages that could have been used to instantiate the proposed framework.

The processing of the data is performed by Matlab code. The DESI 1.0 interface is linked to a MySQL database where all the data are stored. The interface can be made available to any user via the Matlab Compiler, even if the end user’s computer does not have Matlab [37].

The flow of information from **Figure 4** is presented in **Figure 6**, but with the actual tools used in the development and implementation of DESI 1.0. Data processing starts with the DESI user interface, where users input information in the form of typing, box checking and button pressing. User inputs are processed by the compiled Matlab code and stored in MySQL. Once data are stored in MySQL, the program extracts the data from MySQL using SQL code in order to perform calculations and distribution fitting. Once the data have been processed, DESI 1.0 sends the relevant information back to the analyst in the form of message boxes and relevant graphics.

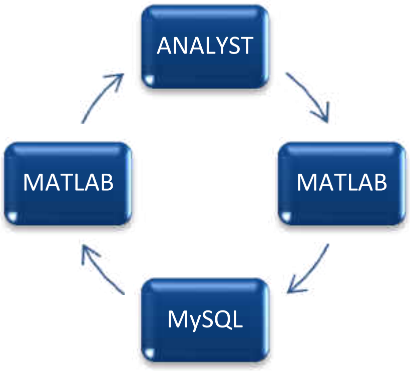
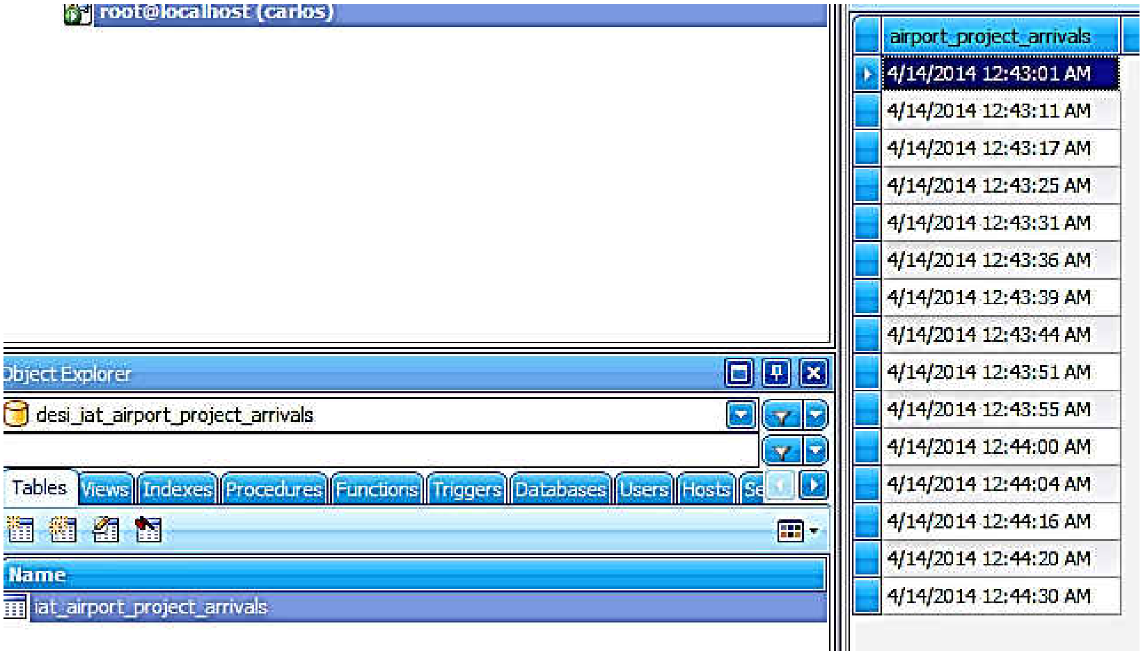


Figure 6 Flow of information after Matlab and MySQL implementation in DESI 1.0.

Blue Section of DESI 1.0

The top blue section of DESI is used to collect Class (I) data by typing the variable name and pressing the button that says `Press to record data`. The `Press to record data` button activates a Matlab function that generates SQL statements to produce the current timestamp for entity *i*, t_i. When entity *i+1* arrives, the pressing of `Press to record data` records t_{i+1} as the timestamp, in the same manner it recorded t_i. Then the DESI logic calculates the positive difference t_{i+1} - t_i in order to calculate interarrival time IAT$_i$. Arrival timestamps remain stored in MySQL in a schema called `DESI_IAT_VN` and within a table called `IAT_VN` where `IAT` stands for interarrival time and `VN` is the variable name. The number of rows of table `DESI_IAT_VN.IAT_VN` is equal to the number of collected arrival times minus one. Calculated interarrival times are also exported to a text file in case the user needs to load them into another application. Data files are named `DATA_VN_interarrival_times.txt`, where `VN` stands for variable name.

For example, the timestamps for the variable called `AIRPORT_PROJECT_ARRIVALS`, are saved in the table `DESI_IAT_AIRPORT_PROJECT_ARRIVALS.IAT_AIRPORT_PROJECT_ARRIVALS`, as shown in **Figure 7**. The file exported with the collected data is called `DATA_AIRPORT_PROJECT_ARRIVALS_interarrival_times.txt`.



airport_project_arrivals
4/14/2014 12:43:01 AM
4/14/2014 12:43:11 AM
4/14/2014 12:43:17 AM
4/14/2014 12:43:25 AM
4/14/2014 12:43:31 AM
4/14/2014 12:43:36 AM
4/14/2014 12:43:39 AM
4/14/2014 12:43:44 AM
4/14/2014 12:43:51 AM
4/14/2014 12:43:55 AM
4/14/2014 12:44:00 AM
4/14/2014 12:44:04 AM
4/14/2014 12:44:16 AM
4/14/2014 12:44:20 AM
4/14/2014 12:44:30 AM

Figure 7 The loading and storage of arrivals in MySQL.

After a new timestamp is inserted into the table `DESI_IAT_VN.IAT_VN`, the Matlab code does the following:

1) Extracts the data from `IAT_VN` performing the query "`SELECT * FROM DESI_IAT_VN.IAT_VN`" so it can be processed using Matlab code; for the simple example shown in **Figure 7** this query becomes

   ```
   SELECT * FROM DESI_IAT_AIRPORT_PROJECT_ARRIVALS.IAT_
   AIRPORT_PROJECT_ARRIVALS
   ```

2) Using Matlab code, consecutive arrival times are subtracted as follows

   $$(Interarrivaltime)_i = t_i\text{-}t_{i-1} \tag{4.1}$$

   where t_i stands for the i^{th} collected arrival time. The calculated interarrival time is then converted into a continuous variable and stored in a vector of continuous data for probability distribution fitting. The time difference is converted to minutes using formula (4.2) for entry *i*, *i*=1,…,N

   $$\text{Entry } i\text{=DD•24•60+HH•60 + MM + SS/60} \tag{4.2}$$

   …where DD represents the days, HH represents the hours, MM represents the minutes and SS represents the seconds. So for example, if interarrival time *i*, e.g. IAT_i, is 2 minutes 30 seconds, it is converted to 2.5 minutes.

3) Once the time differences are calculated using (4.1) and converted into a continuous variable using (4.2), interarrival times are arranged into a histogram. The histogram updates after each collected arrival time.

4) The resulting best fit is obtained using the rules detailed in chapter three. The best fit is displayed with the corresponding histogram as shown in **Figure 8**. Histograms and best fitted distributions update and display in real-time after a new timestamp is added.

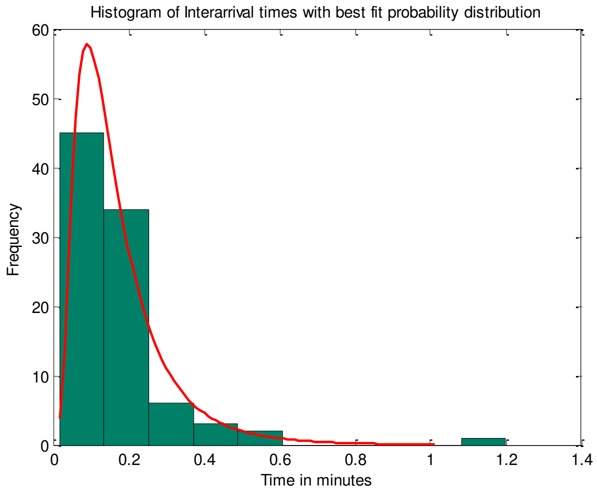
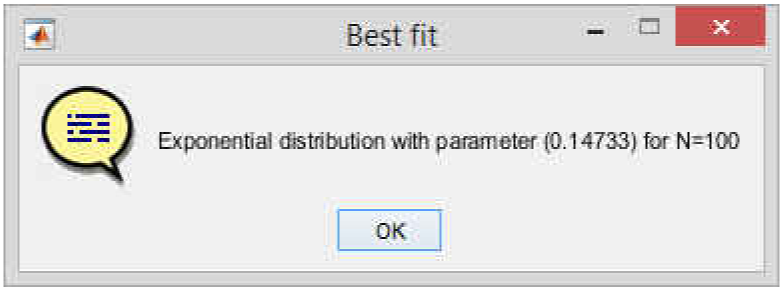


Figure 8 Histogram and best fit display

For both versions of DESI, the default number of histograms bins is the square root of the number of entries in the data, rounded up. Nevertheless, both versions of DESI offer the option to set the number of histogram bins. The resulting best fit distribution and its parameters are displayed in a message box like the one shown in **Figure 9** so the analyst can enter this information in the DES software; or the analyst can keep recording data by pressing the button `Press to record data`, which will likely produce a different best fit each time.

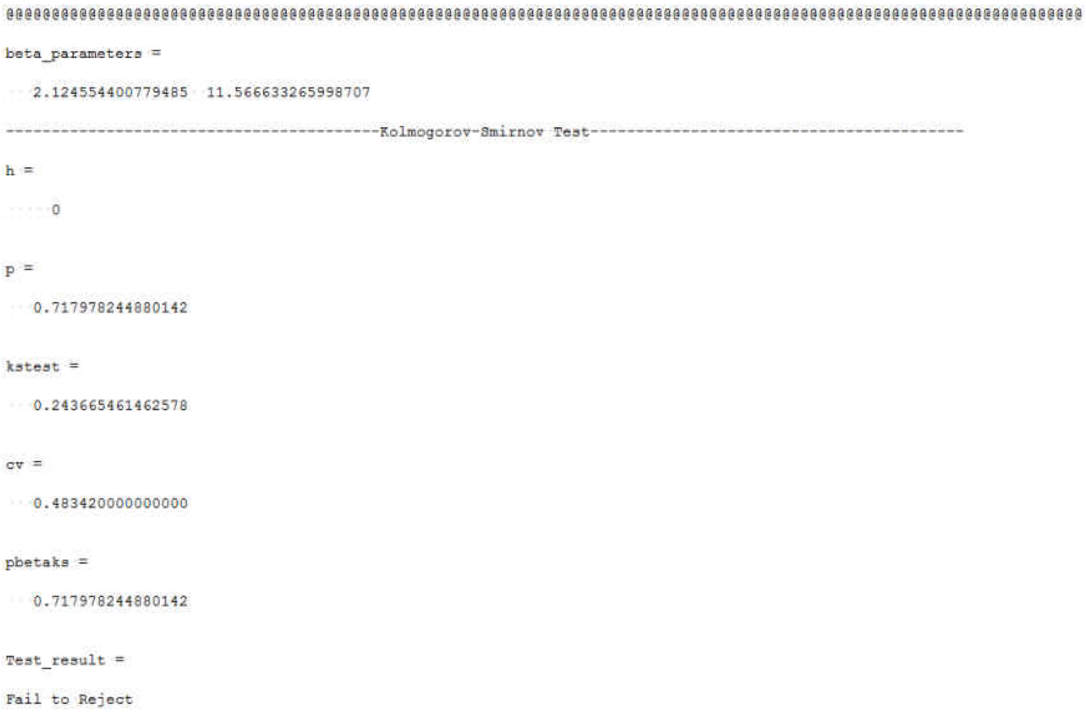


Exponential distribution with parameter (0.14733) for N=100

Figure 9 Best fit distribution after 100 observations.

When DESI 1.0 is run for the first time, MySQL database connection parameters need to be defined. The first run of DESI 1.0 creates a folder called `C:\DESI`, where a text file of connection parameters is created. This critical file is called dbconnection2.txt. More details on dbconnection2.txt are presented later in the document. When DESI is run for the first time, a log folder is also created, namely C:\DESI\LOG,where DESI 1.0 saves text files with collected data, goodness-of-fit tests results and simulated data if this option is selected.

Figure 10 shows an example of the Kolmogorov - Smirnov goodness-of-fit test results for a Beta distribution fit saved in C:\DESI\LOG. Test results are provided for each of the selected probability distributions and updated right after new data are collected. In general, the goodness-of-fit text file is named GOF_VN_interarrival_times.txt, where VN stands for variable name. For the hypothetical variable shown in **Figure 7** named AIRPORT_PROJECT_ARRIVALS, the file generated with the goodness-of-fit test results is GOF_AIRPORT_PROJECT_ARRIVALS_interarrival_times.txt.



```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

beta_parameters =

   2.124554400779485   11.566633265998707

-----------------------------------------Kolmogorov-Smirnov Test-----------------------------------------

h =

     0


p =

   0.717978244880142


kstest =

   0.243665461462578


cv =

   0.483420000000000


pbetaks =

   0.717978244880142


Test_result =

Fail to Reject
```

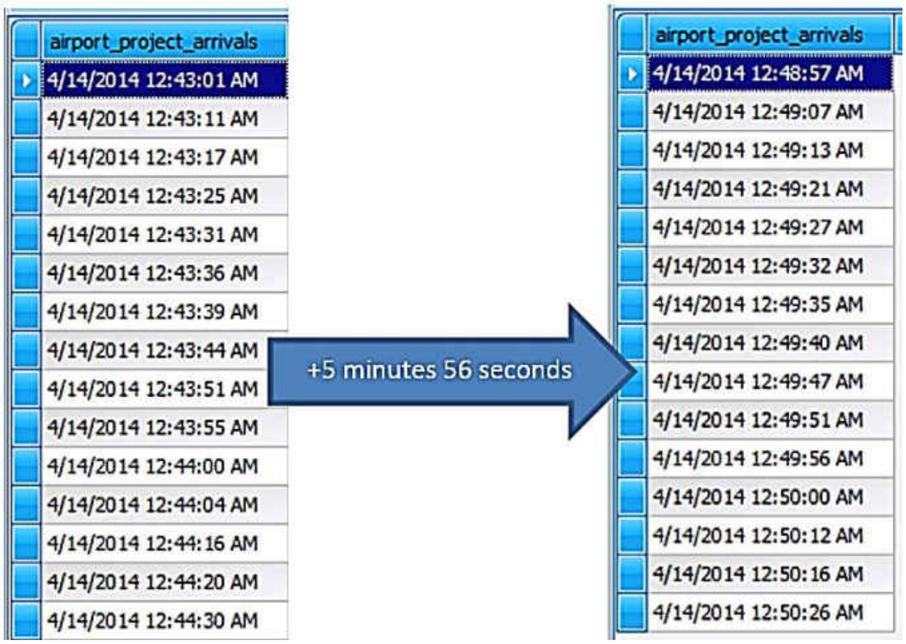
Figure 10 Partial view of goodness of fit test results.

After each collected observation, previously generated text files are overwritten with the updated information. DESI 2.0, discussed later in this document, produces the same data, but skips the MySQL part.

As detailed in chapter three, users also have the option to resume the collection of interarrival times at a later time. In the blue section of DESI, this is accomplished by pressing the button labeled "`Press to resume a previous collection. Press when first arrival occurs in the new collection period`" (PRPC). When PRPC is pressed, DESI understands that the last value from the previous collection is not the first value of the resumed collection. In the proposed logic, the last value from the previous collection $t_{N,\text{original}}$ (N means the latest one from a sorted list of N elements) artificially becomes the first value of the resumed collection $t_{1,\text{resumed}}$ by converting $t_{N,\text{original}}$ into the exact timestamp of the resumed collection. Then the logic converts all the previously recorded values by adding to all of them, the difference between $t_{N,\text{original}}$ and $t_{1,\text{resumed}}$, e.g. $\sigma_{(i+1)\vartheta}$ from eq. (3.11). In other words, all the arrival times from the original collection get modified by $\sigma_{(i+1)\vartheta}$, but the interarrival times remain the same, preventing corruption in the calculated data.

The `PRPC` button in the blue section of DESI must be pressed when the first arrival of the resumed collection occurs. Then timestamp $t_{N,\text{original}}$ gets converted into the resuming effort timestamp and represents the first arrival of the resumed collection, $t_{1,\text{resumed}}$. Then when the second entity arrives in the resumed collection at $t_{2,\text{resumed}}$, the difference between these arrivals ($t_{2,\text{resumed}} - t_{1,\text{resumed}}$) becomes the first interarrival time of the resumed collection. Interarrival times from a resumed collection are then appended to the interarrival times of the original collection. Then users can continue recording Class (I) data in the usual manner.

The following example illustrates the logic for resuming a previous collection of Class (I) data in the blue section of DESI. For simplicity, let us use different timestamps of the same day. The last arrival in the original collection from **Figure 7** occurred at $t_{N,\text{original}}$ = 12:44:30 AM. Let say the user wants to resume the collection at 12:50:26 AM. The difference between these two timestamps is 5 minutes and 56 seconds, hence $\sigma_{2,\ AIRPORT\_PROJECT\_ARRIVALS}$ = 5.93 minutes, after making it a continuous variable. Then, when the analyst resumes the collection in the second effort, the difference of 5 minutes and 56 seconds is added to every record in the original collection from **Figure 7** in order to preserve the original interarrival times. This is illustrated in **Figure 11**. When the resumed collection happens, $t_{N,\text{original}}$ becomes $t_{1,\text{resumed}}$ = 12:50:26 AM by adding the calculated $\sigma_{2,\ AIRPORT\_PROJECT\_ARRIVALS}$ = 5 minutes 56 seconds to $t_{N,\text{original}}$. That is, 12:44:30 AM + 00:05:56 = 12:50:26 AM. The 5 minutes 56 seconds are also added to every other recorded timestamp in the original collection effort. The difference between $t_{1,\text{resumed}}$ = 12:50:26 AM and $t_{2,\text{resumed}}$ will become the first inter arrival time of the resumed collection effort.



airport_project_arrivals
4/14/2014 12:43:01 AM
4/14/2014 12:43:11 AM
4/14/2014 12:43:17 AM
4/14/2014 12:43:25 AM
4/14/2014 12:43:31 AM
4/14/2014 12:43:36 AM
4/14/2014 12:43:39 AM
4/14/2014 12:43:44 AM
4/14/2014 12:43:51 AM
4/14/2014 12:43:55 AM
4/14/2014 12:44:00 AM
4/14/2014 12:44:04 AM
4/14/2014 12:44:16 AM
4/14/2014 12:44:20 AM
4/14/2014 12:44:30 AM

airport_project_arrivals
4/14/2014 12:48:57 AM
4/14/2014 12:49:07 AM
4/14/2014 12:49:13 AM
4/14/2014 12:49:21 AM
4/14/2014 12:49:27 AM
4/14/2014 12:49:32 AM
4/14/2014 12:49:35 AM
4/14/2014 12:49:40 AM
4/14/2014 12:49:47 AM
4/14/2014 12:49:51 AM
4/14/2014 12:49:56 AM
4/14/2014 12:50:00 AM
4/14/2014 12:50:12 AM
4/14/2014 12:50:16 AM
4/14/2014 12:50:26 AM

Figure 11 Logic to resume a previous collection and preserve interarrival times

Then the analyst can continue recording interarrival times as usual by pressing the button labeled `Press to record data`. The logic for resuming a previous Class (I) data collection is an original contribution of this research work to the field of modeling and simulation.

The blue section of DESI has the option to display, fit, test and validate previously collected Class (I) data at any time. This is accomplished by pressing the button labeled `Press to display collected data`, which does not add new values to the list of interarrival times, just displays what has been already collected. This feature is consistent with the ability to verify the data at any time found in methodologies (A) and (B) solutions [8].

The blue section of DESI also permits deleting the last entry recorded by pressing the button labeled `Delete last entry`. DESI 1.0 can delete the entire set of collected data values in MySQL by pressing the button labeled `Start Over`. If the user chooses to start over, the data will only be deleted in MySQL but preserved in the log folder as a backup. In DESI 2.0, the deletion removes the text data files, but saves a copy of the data files with the prefix "Backup_". Autocorrelation diagnostics, multimodal analysis and data simulation can be activated in the blue section of DESI by checking the appropriate boxes of the interface.

Orange Section of DESI 1.0

The middle orange section of the DESI user interface collects Class (II) data by typing the variable name and by pressing the buttons labeled `RECORD START` and `RECORD END` in order to collect the starting timestamp and ending timestamp of a specific event. Right under the `RECORD START` and `RECORD END` buttons is the option to delete the last entry or to delete the entire set of entries for the variable being recorded. After entering the variable name (`VN`), the

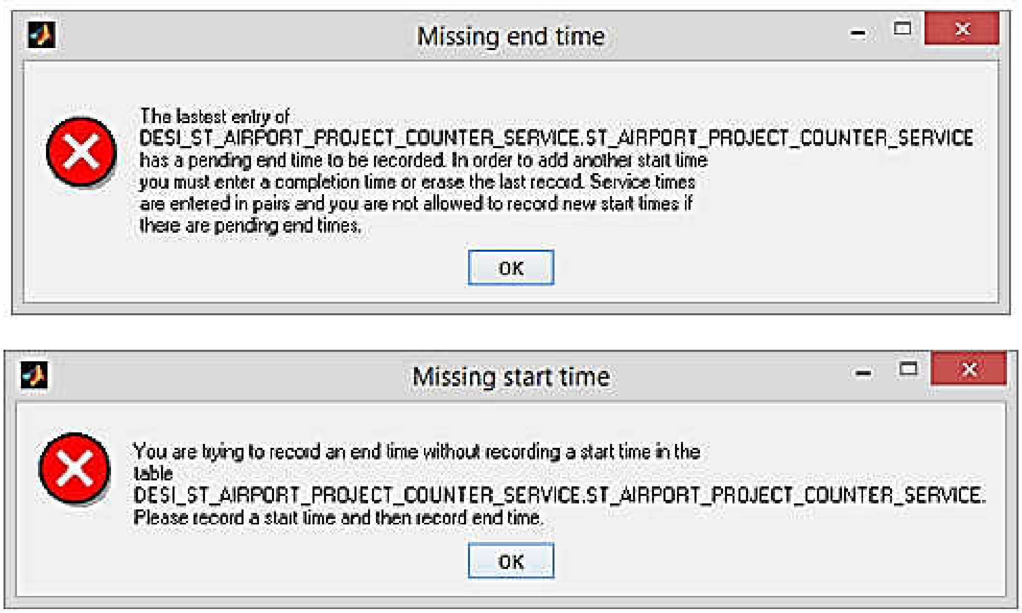
RECORD START button activates a compiled Matlab function that generates and stores the current timestamp for entity i, $t_{i,\text{start}}$. In the same manner, the RECORD END button generates and stores another timestamp for entity i, $t_{i,\text{end}}$. Then the underlying programming of DESI 1.0 subtracts $t_{i,\text{end}}$ - $t_{i,\text{start}}$ in order to calculate service time ST_i. Timestamps are stored within a MySQL table called DESI_ST_VN.ST_VN where ST stands for service time and VN is the variable name. The number of rows of DESI_ST_VN.ST_VN is equal to the number of start timestamps. Collected data values and test results are saved in the log folder with filenames of the form DATA_VN_service_times.txt and GOF_VN_service_times.txt, where VN stands for variable name. For example, the variable COUNTER_SERVICE is stored in the table DESI_ST_COUNTER_SERVICE.ST_COUNTER_SERVICE.

After the data are stored in MySQL, the processing is similar to the processing of the blue section. That is, the data gets converted into a continuous variable, displayed in a histogram, fitted, tested and validated. Feedback is provided to the analyst in the form of messages and images. Then the analyst decides whether or not to keep adding data that generate different fits. As new data are added, the histograms and fitted distributions get updated with the new information. If the analyst decides not to add more data values, then the DESI 1.0 output can be added to DES software. Autocorrelation diagnostics, multimodal analysis and data simulation are also available in the orange section of DESI by checking the appropriate boxes of the interface.

Resuming the collection in the orange section is straight forward and processing is much simpler than doing so in the blue section because for each collection there will always be a start time and an end time, regardless of the time of the collection. The orange section is constrained [22] by the following conditions

- No end time can be recorded if there is no associated start time (i.e. it is not allowed to collect two consecutive end times).
- Two consecutive start times cannot be recorded because a start time must be followed by its corresponding end time.

Users receive an error message if there is an attempt to violate any of these constraints. **Figure 12** shows these notifications. The orange section of DESI 2.0 performs the same functionalities, except that it does not save data in MySQL tables but in text files, which are named using the variable names as prefix. In DESI 2.0, the deleted variables are backed up using the prefix "Backup_" in the filename.



Missing end time

The lastest entry of
DESI_ST_AIRPORT_PROJECT_COUNTER_SERVICE.ST_AIRPORT_PROJECT_COUNTER_SERVICE
has a pending end time to be recorded. In order to add another start time
you must enter a completion time or erase the last record. Service times
are entered in pairs and you are not allowed to record new start times if
there are pending end times.

Missing start time

You are trying to record an end time without recording a start time in the
table
DESI_ST_AIRPORT_PROJECT_COUNTER_SERVICE.ST_AIRPORT_PROJECT_COUNTER_SERVICE.
Please record a start time and then record end time.

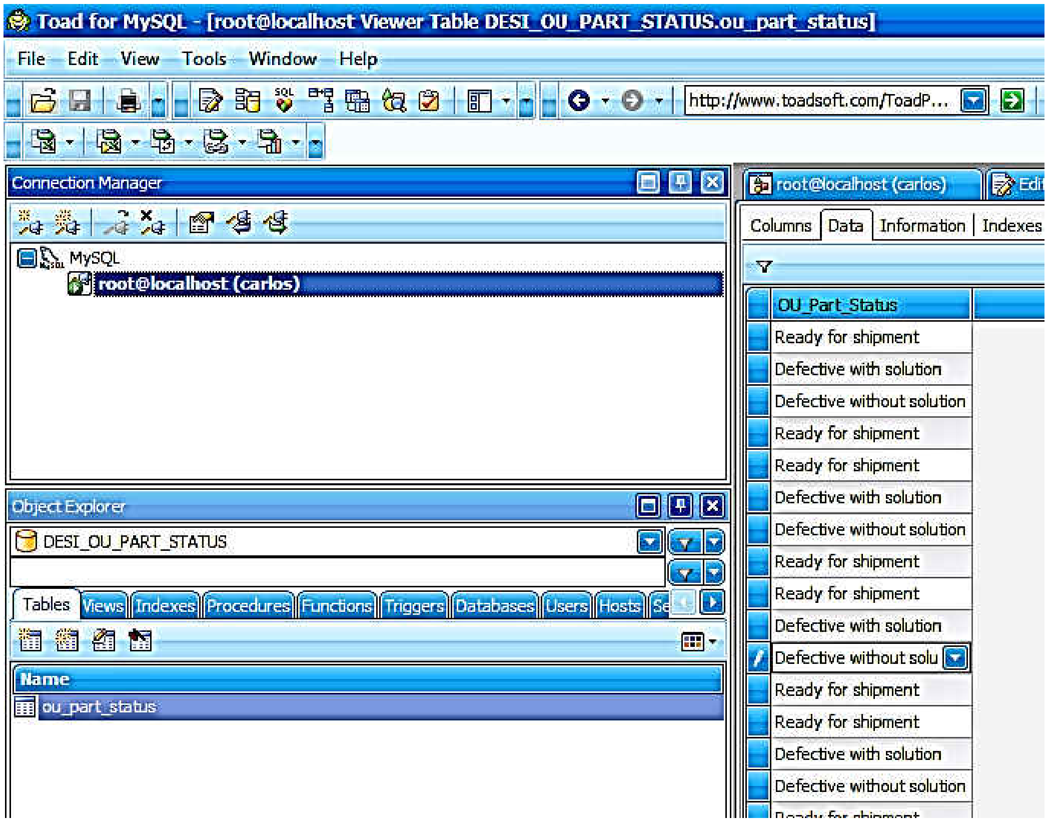
Figure 12 Constraints in the orange section

Green Section of DESI 1.0

The bottom green section of DESI collects Class (III) data. The analyst defines the variable name and also defines each of the possible outcomes of that variable at a given decision point within the modeled system. DESI allows up to 10 possible outcomes per variable at a given decision point. Within the green section of the DESI user interface, there are buttons labeled `Press to tally outcome` that tally the occurrence of the outcome specified above it by typing the name of the potential outcome.

In DESI 1.0, after pressing `Press to tally outcome`, the specified outcome of the variable `VN` is inserted into a MySQL table called `DESI_OU_VN.OU_VN`, where `OU` stands for outcome and `VN` stands for variable name. After a new value is tallied and inserted into `DESI_OU_VN.OU_VN`, a pie chart is produced detailing the relative frequencies or probabilities for each outcome. This pie chart contains information that can be entered into DES software at specific decision points within the modeled system, as shown in **Figure 3**. Each piece of the pie chart represents the γ_{im}'s presented in chapter three. The γ_{im}'s are added to DES software [4] in order to direct the flow of entities through the simulated system.

A partial view of the MySQL table that stores the data from **Figure 3** is shown in **Figure 13**. Collected Class (III) data are also saved as text files in the log folder `C:\DESI\LOG`, with filenames of the form `TALLY_VN_outcomes.txt`, where `VN` stands for variable name. For example, the text file that contains the collected data for the variable `PART_STATUS` shown in **Figure 3** takes the name `TALLY_PART_STATUS_outcomes.txt`.



Toad for MySQL - [root@localhost Viewer Table DESI_OU_PART_STATUS.ou_part_status]

Name
ou_part_status

OU_Part_Status
Ready for shipment
Defective with solution
Defective without solution
Ready for shipment
Ready for shipment
Defective with solution
Defective without solution
Ready for shipment
Ready for shipment
Defective with solution
Defective without solu
Ready for shipment
Ready for shipment
Defective with solution
Defective without solution
Ready for shipment

Figure 13 Class (III) data recorded using the green section of DESI 1.0.

Like the other two sections of DESI, the green section can display previously collected data, delete the latest recorded entry or delete all the observations for a given variable. The green section of DESI 2.0 performs the same functionalities as the green section of DESI 1.0, with the difference that the data are not stored in MySQL, but in text files.

In addition to the aforementioned features, the three sections of DESI respond to the activation of optional parameters presented later in this document.

Implementation of DESI 1.0

In order to make DESI more attractive than "difficult" methodology (C) and methodology (D) solutions [8], the implementation of DESI must be simple.

When DESI 1.0 is run for the first time, it creates the folders `C:\DESI` and `C:\DESI\LOG` and the file called `C:\DESI\dbconnection2.txt`. DESI 1.0 notifies the user to enter the right connection parameters in the file `dbconnecton2.txt`. These parameters must be entered and saved the very first time DESI 1.0 is utilized. The location of `dbconnection2.txt` and the folders created by DESI 1.0 are shown in **Figure 14**.

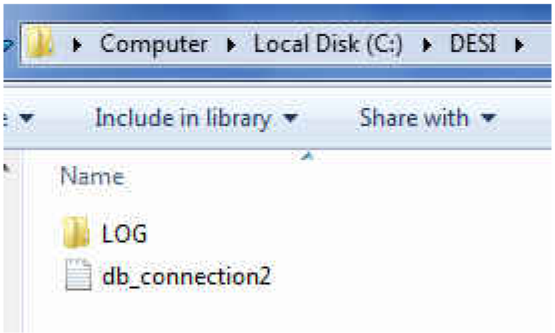
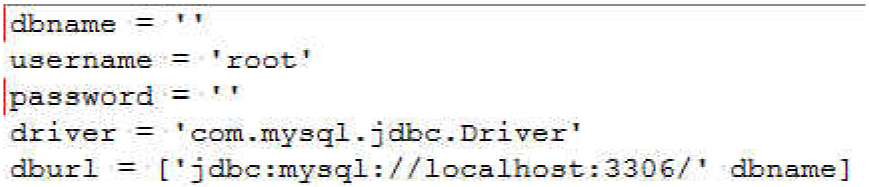


Figure 14 Parameters file and log folder created when DESI 1.0 is run the first time.

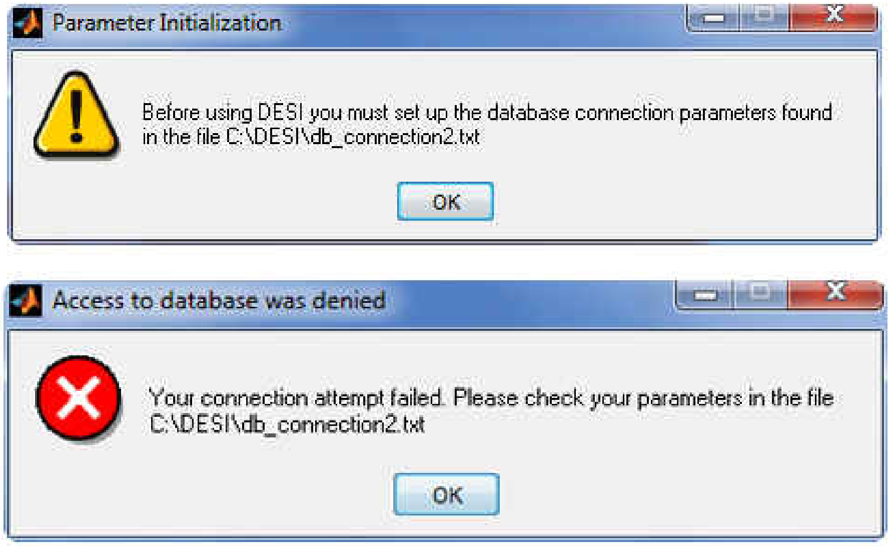
Figure 15 shows the content of `dbconnection2.txt`. The connection parameters' setup is the only implementation requirement for DESI 1.0 beyond the installation wizard.



```
dbname = ''
username = 'root'
password = ''
driver = 'com.mysql.jdbc.Driver'
dburl = ['jdbc:mysql://localhost:3306/' dbname]
```

Figure 15 Content of parameters' file `dbconnection2.txt`.

In general, within `dbconnection2.txt`, lines 2, 4 and 5 will not need to be changed most of the time. Only `dbname` and `password` parameters will need to be modified during the implementation of DESI 1.0. The parameter `dbname` must be defined as any schema name present in MySQL. The default schema name is `sakila`, which is defined during the installation of MySQL, and can be used in the `dbconnection2.txt` file (so `dbname= 'sakila'`). Or the user can create a new schema within MySQL and use that one instead. It does not really matter what schema name is used, as long as it exists in MySQL. This is only for connection purposes and will not affect DESI 1.0 processing. DESI 1.0 eventually creates its own schemas, but installation requires one to exist initially, so the connection can be created [38 - 41]. The password parameter must match the password created in MySQL during installation. DESI 1.0 notifies the user whether the connection parameters need to be initialized or reentered. **Figure 16** shows these scenarios.



Parameter Initialization

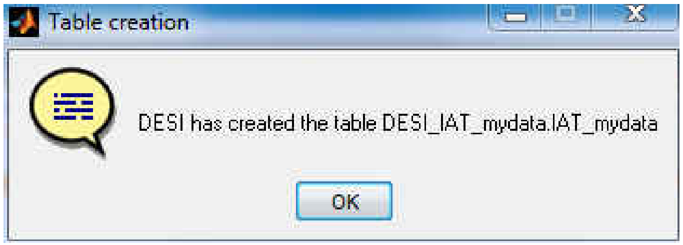
Before using DESI you must set up the database connection parameters found in the file C:\DESI\db_connection2.txt

Access to database was denied

Your connection attempt failed. Please check your parameters in the file C:\DESI\db_connection2.txt

Figure 16 Messages to set up the database connection.

If the connection to MySQL is successful, DESI 1.0 creates the necessary tables and a message is sent to the user, as shown in **Figure 17**. After the successful connection, there is no need to modify `dbconnection2.txt` anymore unless the connection parameters are modified.



DESI has created the table DESI_IAT_mydata.IAT_mydata

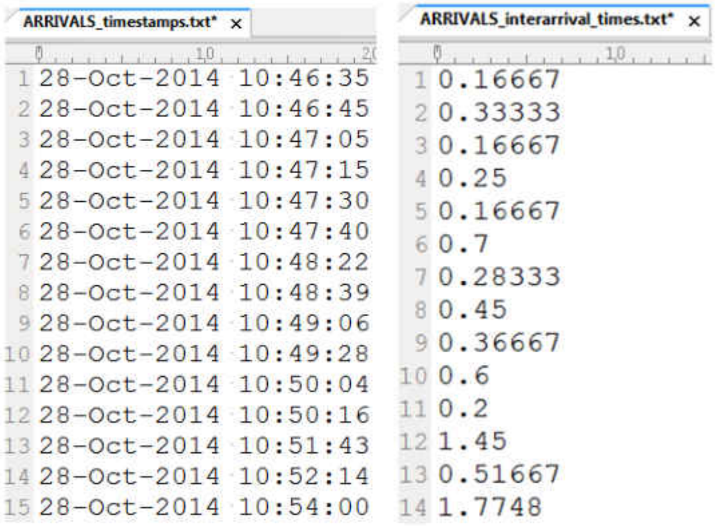
Figure 17 Successful connection to MySQL.

Once these parameters have been added and saved, DESI 1.0 is ready to be used. In summary, the implementation of DESI 1.0 only needs the installation of MySQL, the installation of DESI 1.0 following the installation wizard and the definition of the parameters in `dbconnection2.txt`.

Development of DESI 2.0

The second version of DESI, called DESI 2.0, does not rely on database connectivity, which makes its implementation simpler than DESI 1.0. Even though DESI 1.0 and DESI 2.0 are equivalent in many respects, DESI 2.0 simplifies the implementation process further because there is no need to set up database connections. DESI 2.0 presents important developments to DESI 1.0. For instance, DESI 2.0 can be installed on mac and pcs, whereas DESI 1.0 only works on pcs and DESI 2.0 has the option for speed calculation in the orange section. Nevertheless, the general features in DESI 2.0 are equivalent to those found in DESI 1.0. For instance, the three sections of DESI 1.0 have the same functionality in DESI 2.0.

In DESI 2.0, the data are stored in text files and not in MySQL tables. The data files are produced and saved in the folder `C:\DESI\DATA\` if running DESI 2.0 on a pc or in the folder `/Applications/DESI/DATA/` if running DESI 2.0 on a mac. The name of the collected variable is the prefix of the corresponding data filename so the user can locate the data within the folder `C:\DESI\DATA\` if running DESI 2.0 on a pc or in the folder `/Applications/DESI/DATA/` if running DESI 2.0 on a mac. **Figure 18** shows the collected data and the calculated data in text files using DESI 2.0 for the hypothetical variable called ARRIVALS. The timestamp information contained in **Figure 18** is equivalent to the information contained in **Figure 7**, with the difference that **Figure 7** displays the data stored in MySQL generated using DESI 1.0.



ARRIVALS_timestamps.txt*

```
28-Oct-2014 10:46:35
28-Oct-2014 10:46:45
28-Oct-2014 10:47:05
28-Oct-2014 10:47:15
28-Oct-2014 10:47:30
28-Oct-2014 10:47:40
28-Oct-2014 10:48:22
28-Oct-2014 10:48:39
28-Oct-2014 10:49:06
28-Oct-2014 10:49:28
28-Oct-2014 10:50:04
28-Oct-2014 10:50:16
28-Oct-2014 10:51:43
28-Oct-2014 10:52:14
28-Oct-2014 10:54:00
```

ARRIVALS_interarrival_times.txt*

```
0.16667
0.33333
0.16667
0.25
0.16667
0.7
0.28333
0.45
0.36667
0.6
0.2
1.45
0.51667
1.7748
```

Figure 18 Data storage using DESI 2.0

DESI 2.0 has eight global parameters. These parameters surround the main sections of the user interface, e.g. blue, green and orange. Most of these parameters are also found in DESI 1.0, except for the speed calculation feature. Leaving the spaces for parameters blank or unchecked activates the default values. The parameters to customize the user experience using DESI are

1) The level of significance α (default value is 0.05, and $0< \alpha <1$);

2) The moving window of observations (default is the total number of observations);
3) The purge date, so any data collected prior to that date gets deleted;
4) The number of histogram bins (default is the square root of the number of elements in the data, rounded up);
5) The cutoff values for fitting multimodal data (default fits all data values);
6) The lag parameter for autocorrelation visual diagnostics (default is 1);
7) The number of new data values to be simulated based on resulting probability distribution fitting (default is 100).
8) The distance for speed calculations[2].

The moving window of observations only fits the latest $h \leq N$ collected observations. For example, if there are N=100 collected data values, but the moving window parameter is 75, only the last h=75 collected values out of the 100 observations are analyzed by DESI. The purge date parameter must be entered using the format `YYYY-MM-DD` or `YYYY-MM-DD HH:MM:SS` where `HH` is in 24-hour format (i.e. 00 to 23). If the wrong date format is used or if the user tries to purge data without entering any dates, an error message is displayed, as suggested by the "feedback" HCI design principle [35]. The fourth optional parameter offers the possibility to create histograms with different number of bins. The reason for offering this option is that sometimes the shape of the histogram can impact the best fit probability distributions [4].

The fifth optional parameter offers the capability to fit subsets of the data. It applies only to the blue and orange sections. This optional parameter seeks to handle multimodal data [4] and if enabled, DESI fits the data between the cutoff values, reports the percentage of processed records out of the full set of data values and produces histograms of the data that falls outside the

[2] Feature only available in DESI 2.0.

selected cutoff values. The multimodal parameter works in conjunction with checking the multimodal check box. The sixth optional parameter is the lag parameter, which works in conjunction with checking the box for autocorrelation diagnostics. This optional parameter works in the blue and orange sections of DESI. Autocorrelation diagnostics are important in DES because independence among collected observations is a key building block for different theoretical results such as the maximum likelihood estimation of probability distribution parameters and goodness-of-fit tests [3].

The seventh parameter represents the number of data values to be simulated and it works in conjunction with checking the box for the simulation of new data. It applies to the blue and orange sections of DESI. When the simulation option is enabled, DESI produces simulated data and saves them as text files, which are ready to be imported into DES software.

The eighth parameter is the distance parameter, which is used to calculate speeds given Class (II) data associated with transport times. This parameter, when checked, only applies to the orange section of DESI.

Calculation of Speed in the Orange Section of DESI 2.0

DESI 2.0 permits the calculation of speed in miles per hour from the data collected in the orange section as presented in eq. (3.12). The calculation of speed only requires checking the speed calculation checkbox and entering the distance associated with the transport in meters. Distance is part of the model logic, when applicable. Transport times are calculated using the orange section in the usual manner, where the start timestamp is the departure time and the end timestamp is arrival time. The visual output is a fit of the calculated speeds plus the relevant descriptive statistics so DES modelers can enter this information in DES software. When the speed parameters are entered, the fitting is on the speeds calculated from the collected Class (II) data and not on the collected Class (II) data per se. The calculated speeds are a function of the Class (II) data, hence there is no need for a special data class for speed data as these fall under the umbrella of Class (II) data.

Optional Autocorrelation Diagnostics

For the data collected using the blue and orange sections of DESI there is the option to produce plots for the visual inspection of autocorrelation [3, 42, 43] for different lags of the collected variable. This feature is useful in confirming the random nature of the collected observations, which is an important assumption in DES theory [3]. Users need to check the box that activates the autocorrelation diagnostic plots and also need to provide the number of lags to be analyzed (if left blank, the default value is 1 period lag). Checking the box for autocorrelation diagnostics provides:

- The correlation coefficient between *X(t)* vs *X(t-j)* where *j* is the selected number of lags;
- Statistical testing for H_0: no autocorrelation vs H_1: autocorrelation;
- A scatter plot of *X(t)* vs *X(t-j)* where *j* is the selected number of lags;
- A correlogram;
- A plot of the series with the mean of the series; and
- A plot of the calculated data versus the timestamp of the calculated data[3].

[3] Feature only available in DESI 2.0.

Figure 19 shows the output of selecting this option for a hypothetical variable called `INTERARRIVAL` and a lag of 1. **Figure 19** shows the resulting lag correlation coefficient and whether or not it is significant at the selected level of significance, e.g. α from the global parameters section. The bottom right corner of **Figure 19** presents the scatter plot of the calculated times by the timestamps of the data calculations. This plot includes the least squares regression line thru the scatter plot intending to reflect a potential relationship between the calculated data and the timestamps. A statistically significant slope coefficient indicates that the time of the data entry can explain the calculated data [43]. The relevance of a statistically significant slope coefficient, based on the t-test [43], varies depending on the type of DES model being carried out. The purpose of producing the plot of calculated data by timestamps is to show the times of the day when there is more activity in order to produce more realistic simulations. In this plot, the timestamps range from 00:00 to 23:59, or equivalently from 00.00 to 23.99 when converted to a continuous variable. The scatter plot of calculated times by timestamps is unaffected by the timestamp adjustments when resuming a previous Class (I) data collection.

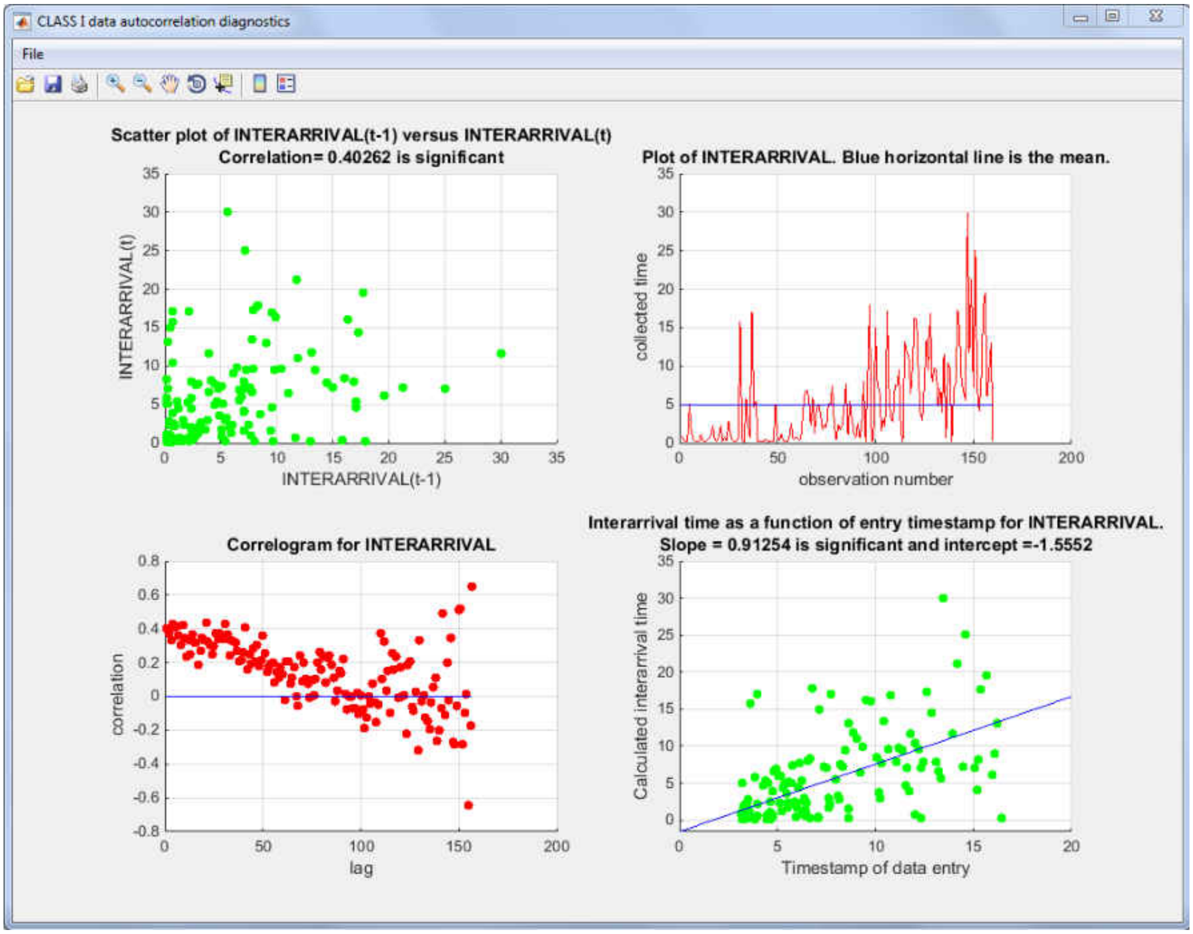

CLASS I data autocorrelation diagnostics
File
Scatter plot of INTERARRIVAL(t-1) versus INTERARRIVAL(t)
Correlation= 0.40262 is significant
INTERARRIVAL(t)
INTERARRIVAL(t-1)
0
5
10
15
20
25
30
35
Plot of INTERARRIVAL. Blue horizontal line is the mean.
collected time
observation number
50
100
150
200
Correlogram for INTERARRIVAL
correlation
lag
0.8
0.6
0.4
0.2
-0.2
-0.4
-0.6
-0.8
Interarrival time as a function of entry timestamp for INTERARRIVAL.
Slope = 0.91254 is significant and intercept =-1.5552
Calculated interarrival time
Timestamp of data entry

Figure 19 Autocorrelation diagnostic plots.

Simulated Data for Automated Import into DES Software

If the boxes for simulated data are checked in either version of DESI, the blue and orange sections create text files of data that are ready to be imported into DES software, if the DES software has that capability. In other words, new data can be simulated based on best fit, triangular fit or uniform fit and passed directly to the DES software so simulations can be performed. This is an alternative to entering the probability distribution into the DES software for data generation. Resulting files with simulated data are saved in the log folder with the prefix

`SIMULATED_`, `TRIANGULAR_` or `UNIFORM_` depending on the boxes checked in the interface. **Figure 20** shows these options within the DESI interface.

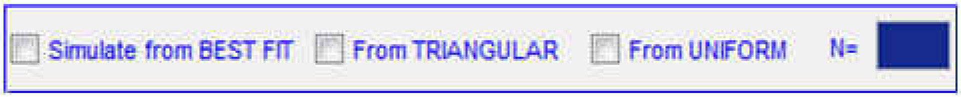


Figure 20 Options for simulating new data and feed DES software.

The reason for having specific triangular and uniform options is the fact that when data are not available, triangular and uniform are viable options to approximate distributions [4]. DESI 1.0 and DESI 2.0 let the users determine how many observations to simulate using the *N* parameter; if the *N* box is left blank but one or more of the data simulation boxes are checked in DESI, the default value for *N* is 100.

Even though DESI 2.0 is the evolution of DESI 1.0, both DESI 1.0 and DESI 2.0 instantiate the proposed framework for automated DES data collection and processing.

Evaluation of Automation

Based in the "model for types and levels of human interaction with automation" presented by Parasuraman et al. [44], the DESI interface focuses on information acquisition automation and information analysis automation hence the DESI interface presents a "high" rank on these two criteria. The DESI interface does not focus on decision automation or action automation hence DESI presents a "low" rank on these criteria. Ultimately the decision of whether to stop generating data belongs to the simulation analyst. The DESI interface is a mechanism for automated data generation, processing and analysis, which provides simulation-ready information to the simulation analyst, but ultimately it is the analyst who evaluates the output and acts based on it.

Although the following analysis needs to be formally tested on user-experience evaluations, which are the subject of future research, the DESI interface intends to reduce mental workload and boost situation awareness about DES input data. The reduction in mental workload and the boosting of situation awareness are feasible due to the reliability of the DESI interface. Because of the full verifiability of the data when using DESI, complacency [44] is not an issue with the automation obtained from DESI. Nevertheless, using DESI can result in skill degradation [44], since the reliance in DESI may impact the learning of other computer solutions and skills. The cost of misusing the DESI interface can result in misleading simulations studies, nevertheless, if data errors are detected in the data preparation process, the DESI interface allows users to remove unwanted data or start over without major inconveniences.

In summary, Parasuraman et al. [44] made the following remark about data automation, which applies directly to DESI: "Certainly cumbersome and clumsy data entry remains a viable candidate for automation. But to reiterate the linkage between decision and action automation, if high automation is selected for the latter, then designers should resist the temptation for high automation levels of decision making" [44].

Suggested Training for the DESI Interface

The DESI interface is a novel option for input data management automation, especially when working with unavailable but collectable data from a stochastic, dynamic process or system. The DESI interface instantiates the proposed integrated framework by automating data preparation for Class (I), Class (II) and Class (III) data. In addition, the DESI interface features convenient practical implications such as real-time simulation-ready output, resulting from the aforementioned classification of the data, given the model logic.

The users of the DESI interface should have knowledge of statistical probability theory and familiarity with discrete event simulation modeling in order to fully understand and make the most out of the capabilities of DESI. Such background will ease the learning of the details within DESI. Nevertheless, the implementation and the data generation from pressing buttons can be performed by less trained technical staff, even if these users are not statisticians or engineers.

The suggested training for the DESI interface has been divided in two sets of steps based on the user's technical background. The training includes the following steps:

1) Follow the simple installation wizard steps in order to install DESI on a PC or in a mac.
2) Familiarize users with the differences between Class (I), Class (II) and Class (III) data.
3) Emphasize the three main sections of DESI, e.g. blue, orange and green, and how these relate to Class (I), Class (II) and Class (III) data.
4) Understand how the three data classes are generated from the data generation buttons in the different sections of the DESI interface.
5) Understand how to automatically resume data generation for each of the three data classes using the buttons of the interface.
6) Understand how to retrieve and display previously collected data for each of the three classes using the buttons of the interface.
7) Understand how to delete the last recorded observation and how to delete the entire set of collected data values.

8) Understand how to display and process the last N collected data values.
9) Understand how to purge data before a cutoff date.
10) Understand how to change the number of histogram bins.

Up to step (10), DESI can be used by less trained technical staff, e.g. non-statisticians or non-engineers. After step (10) the users should have knowledge of probability theory and DES modeling in order to better understand the workings of DESI and the results. The remaining steps for training are the following:

11) Go over the probability distributions available in DESI and how to use specific probability distributions.
12) Understand the rules for determining the best fitted probability distributions.
13) Understand where to find the test results and the generated data, in case the user needs to see the details of the goodness-of-fit test results and the collected data values.
14) Understand how the results can be used in DES software, including automated imports of the generated data.
15) Understand how to perform autocorrelation diagnostics and the four diagnostic plots produced by DESI.
16) Understand how to visualize and process multimodal data by entering cutoff values.
17) Understand how to simulate new data based on best fit, triangular fit or uniform fit.
18) Understand how to estimate average speed between two points using the logic of Class (II) data, given the distance.

DESI training should emphasize the benefits of real-time, streamlined data generation and processing, including probability distribution fitting, goodness-of-fit testing and the automatic validation of the IID assumption; these are original and innovative practical results featured by the DESI interface. In addition, DESI training should emphasize the opportunity cost of not using DESI, especially when dealing with unavailable but collectable data and MDCAP.

Commercialization of the DESI Interface

The DESI interface is an original contribution of this research work to the community of DES modelers, especially when simulation analysts must collect and process previously-unavailable input data from a stochastic, dynamic process or system. DESI is especially useful when the data preparation would otherwise be performed semi-manually or using standalone specialized solutions that work in isolation from each other.

As mentioned previously in this manuscript, 80% of DES modelers still rely on manual approaches to input data management; 80% of these indicated that they would like to enjoy the benefits of data processing automation. The reason why this percentage is so high is the fact that DES input data management automation has been regarded as been "too difficult" [8], especially due to the lack of accessibility and the unavailability of internal data sources with simulation-ready data. Such universe of simulation analysts currently preparing data manually or using standalone solutions, are a potential market for the DESI interface.

DESI should also be attractive to university students who must complete DES projects in order to fulfill course requirements. Students' access to internal corporate data is likely restricted, so students completing university projects will, more than likely, have to collect and process DES input data themselves in order to complete course projects. DESI should be able to automate and streamline the data preparation process for undergraduate and graduate students, making them another potential market for the DESI interface.

DESI is ready to be used in simulation studies, as will be presented in chapter 6. Nevertheless, the current appearance and user-friendliness of this tool are subjects of future research. It is my intent to use user experience evaluations in order to make DESI a more attractive product for the aforementioned target markets.

CHAPTER FIVE: COMPARISON BETWEEN AUTOMATED SOLUTIONS

Documented Tools for Automated DES Input Data Management

Earlier in this document it was mentioned that the research on post-simulation data analysis considerably exceeds the research on pre-simulation data preparation [22]. This can explain the small number of documented solutions for automated input data management in DES projects. In other words, there are few tools that can perform different data preparation steps as part of the same streamlined process. It is important to emphasize the fact that there are standalone solutions that effectively perform each of the individual steps needed in the preparation of data for DES projects. Nevertheless, the gap in the field of modeling and simulation consists in the fact that these standalone solutions work in isolation. Such isolated approach to input data management is one of the main reasons for the MDCAP problem.

The literature on input data management for DES projects presents three concrete attempts to automate input data management using a single tool. These documented tools are the GDM-Tool [14], the MDA interface [15] and Parts Data Collection [29]. Figures 20 to 25 display the conceptual approaches and user interfaces used in the development of these tools. **Figure 21** shows the flow of information and implementation requirements of the GDM-Tool. **Figure 22** shows the user interface and different functionalities available in the GDM-Tool user interface. The most challenging step in setting up the GDM-Tool is the customized configuration required in order to fully map the data from internal data systems into simulation-ready output. Configuration requires trained staff and deep knowledge of the underlying data [14].

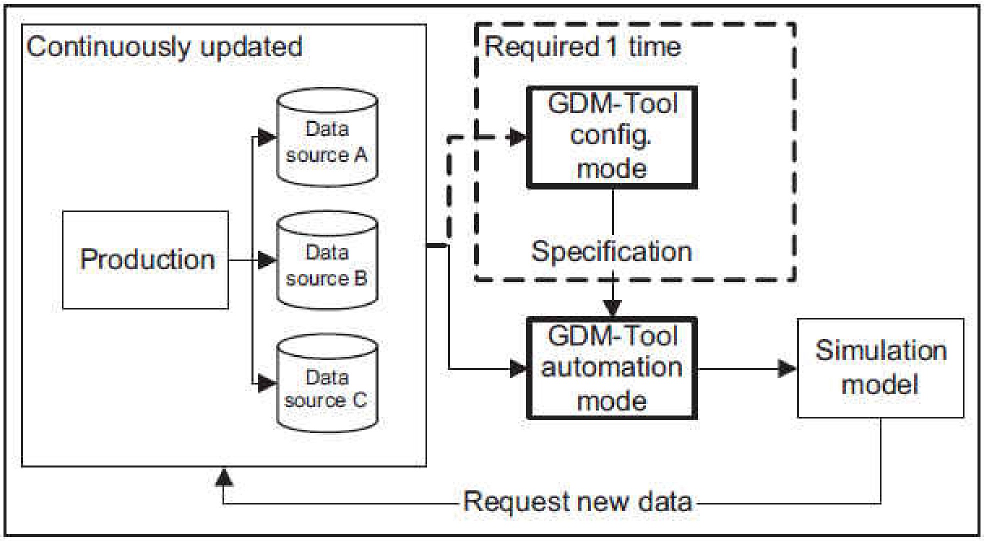


Figure 21 Configuration and Automation of the GDM Tool.

Source: Skoogh, A., Johansson, B., & Stahre, J. (n.d). Automated input data management: evaluation of a concept for reduced time consumption in discrete event simulation. *Simulation-Transactions of the Society for Modeling and Simulation International*, 88(11), 1279-1293.

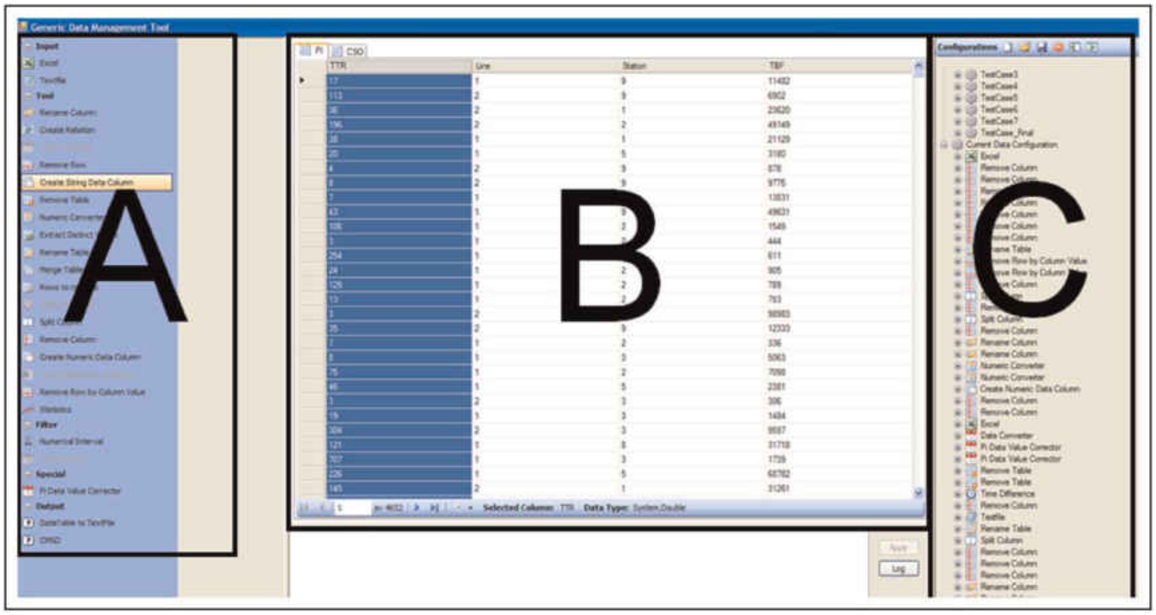
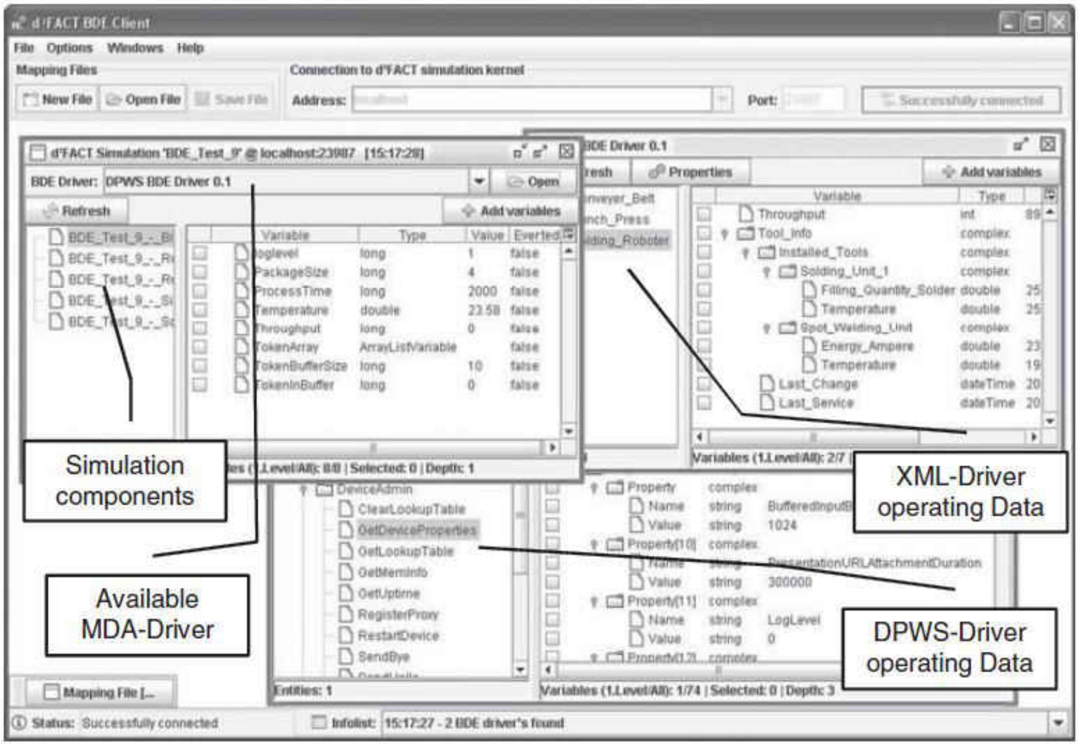


Figure 22 User Interface of the GDM-Tool.

Source: Skoogh, A., Johansson, B., & Stahre, J. (n.d). Automated input data management: evaluation of a concept for reduced time consumption in discrete event simulation. *Simulation-Transactions of the Society for Modeling and Simulation International*, 88(11), 1279-1293.

Figure 23 shows the user interface of the MDA interface. Because MDA retrieves data from internal systems in order to be used in the d3FACT simulator, its applicability is limited. MDA is applicable, on the most part, to manufacturing models and should probably be used by trained technical staff. Familiarity with programming concepts is desirable in order to better understand the workings of this tool. **Figure 24** shows an example of the mapping rules and events used in the MDA interface.



d³FACT Simulation 'BDE_Test_9' @ localhost:23987 [15:17:28]

BDE Driver: DPWS BDE Driver 0.1

BDE_Test_9_-_Bi
BDE_Test_9_-_Ri
BDE_Test_9_-_Ri
BDE_Test_9_-_Si
BDE_Test_9_-_Si

Variable	Type	Value	Everted
loglevel	long	1	false
PackageSize	long	4	false
ProcessTime	long	2000	false
Temperature	double	23.58	false
Throughput	long	0	false
TokenArray	ArrayListVariable		false
TokenBufferSize	long	10	false
TokenInBuffer	long	0	false

...es (1.Level/All): 8/8 | Selected: 0 | Depth: 1

BDE Driver 0.1

...nveyer_Belt
...nch_Press
...lding_Roboter

Variable	Type	
Throughput	int	89
Tool_Info	complex	
Installed_Tools	complex	
Solding_Unit_1	complex	
Filling_Quantity_Solder	double	25
Temperature	double	25
Spot_Welding_Unit	complex	
Energy_Ampere	double	23
Temperature	double	19
Last_Change	dateTime	20
Last_Service	dateTime	20

Variables (1.Level/All): 2/7

DeviceAdmin
ClearLookupTable
GetDeviceProperties
GetLookupTable
GetMemInfo
GetUptime
RegisterProxy
RestartDevice
SendBye

Entities: 1

Property	complex	
Name	string	BufferedInputB
Value	string	1024
Property[10]	complex	
Name	string	PresentationURLAttachmentDuration
Value	string	300000
Property[11]	complex	
Name	string	LogLevel
Value	string	0
Property[12]	complex	

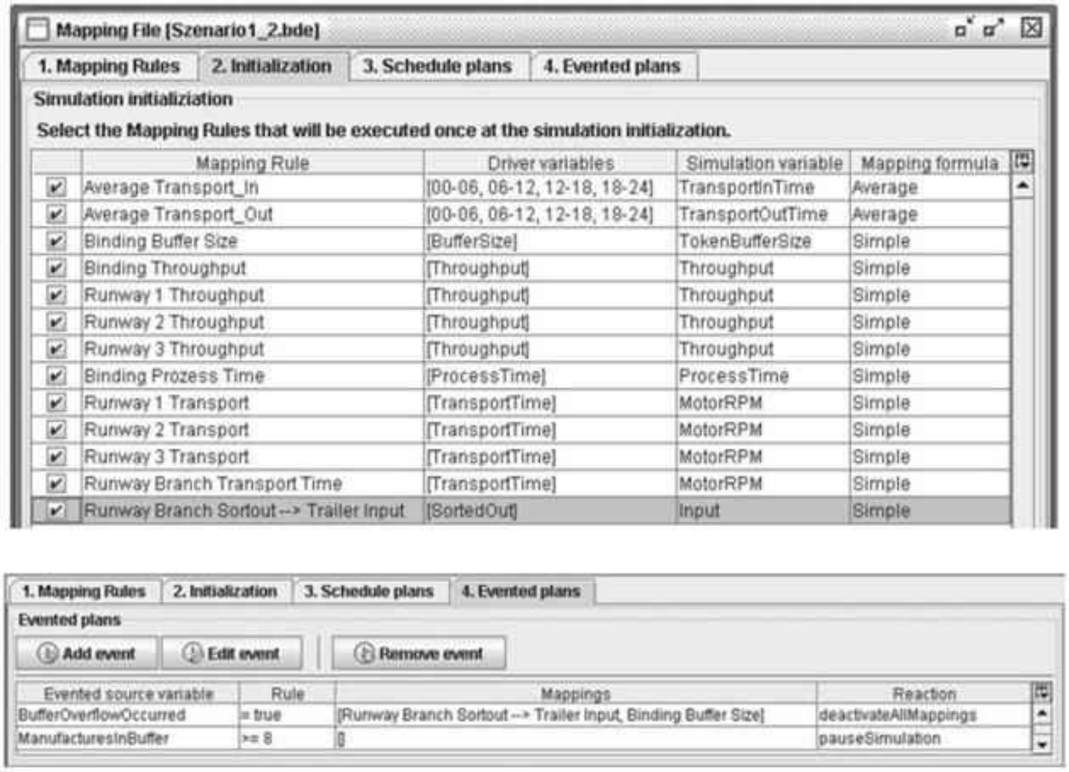
Variables (1.Level/All): 1/74 | Selected: 0 | Depth: 3

Status: Successfully connected

Infolist: 15:17:27 - 2 BDE driver's found

Figure 23 MDA client with opened simulation and MDA driver windows.

Source: Aufenanger, M., Blecken, A., & Laroque, C. (n.d). Design and implementation of an MDA interface for flexible data capturing. *Journal of Simulation*, 4(4), 232-241.



Simulation initializiation

Select the Mapping Rules that will be executed once at the simulation initialization.

	Mapping Rule	Driver variables	Simulation variable	Mapping formula
☑	Average Transport_In	[00-06, 06-12, 12-18, 18-24]	TransportInTime	Average
☑	Average Transport_Out	[00-06, 06-12, 12-18, 18-24]	TransportOutTime	Average
☑	Binding Buffer Size	[BufferSize]	TokenBufferSize	Simple
☑	Binding Throughput	[Throughput]	Throughput	Simple
☑	Runway 1 Throughput	[Throughput]	Throughput	Simple
☑	Runway 2 Throughput	[Throughput]	Throughput	Simple
☑	Runway 3 Throughput	[Throughput]	Throughput	Simple
☑	Binding Prozess Time	[ProcessTime]	ProcessTime	Simple
☑	Runway 1 Transport	[TransportTime]	MotorRPM	Simple
☑	Runway 2 Transport	[TransportTime]	MotorRPM	Simple
☑	Runway 3 Transport	[TransportTime]	MotorRPM	Simple
☑	Runway Branch Transport Time	[TransportTime]	MotorRPM	Simple
☑	Runway Branch Sortout --> Trailer Input	[SortedOut]	Input	Simple

Evented plans

Evented source variable	Rule	Mappings	Reaction
BufferOverflowOccurred	= true	[Runway Branch Sortout --> Trailer Input, Binding Buffer Size]	deactivateAllMappings
ManufacturesInBuffer	>= 8	[]	pauseSimulation

Figure 24 MDA Mapping Rules and Events

Source: Aufenanger, M., Blecken, A., & Laroque, C. (n.d). Design and implementation of an MDA interface for flexible data capturing. *Journal of Simulation*, 4(4), 232-241.

Figure 25 shows the corporate database structure of the Parts Data Collection tool. In this tool, the simulation model receives data from different sources and the data are linked based on a common identifier. This tool focuses on manufacturing applications and was developed for simulation projects within the Lockheed Martin Missiles and Fire Control [29]. **Figure 26** shows the interface sketch and user interface of Parts Data Collection tool. The development of this tool focused on user-friendliness [29] but it is dependent on the existence of data.

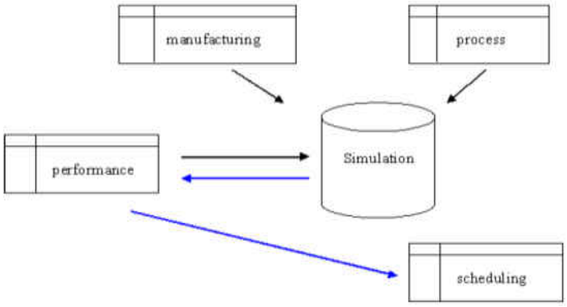


Figure 25 Corporate database structure in Parts Data Collection tool

Source: Portnaya, Irin. (2004). An approach to automating data collection for simulation. (Master Thesis). Retrieved from University of Central Florida Library. (Accession Number ucfl.023120114)

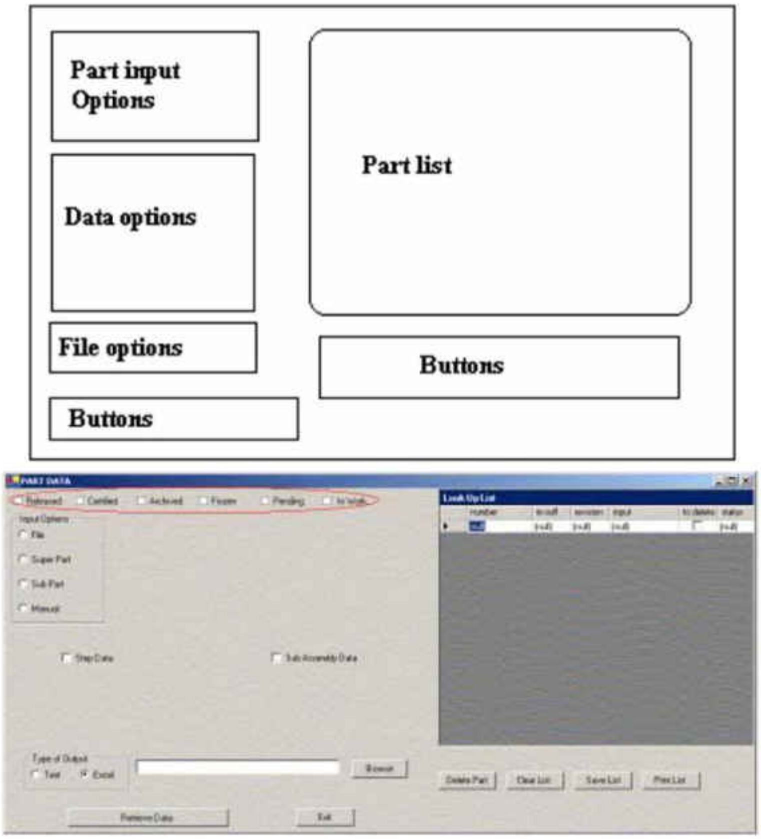


Figure 26 Interface sketch and user interface for Parts Data Collection Tool

Source: Portnaya, Irin. (2004). An approach to automating data collection for simulation. (Master Thesis). Retrieved from University of Central Florida Library. (Accession Number ucfl.023120114)

It has been argued throughout this research investigation that the proposed framework and both versions of the DESI interface are innovative in the field of modeling and simulation and that they advance the state-of-the-art for DES data collection and processing. This chapter focuses on comparing the three documented solutions to the solution offered in this research investigation. The aim of the comparison presented in this chapter is to confirm the uniqueness and originality of the proposed framework and of the DESI interface.

Comparison of Solutions for Automated Input Data Management in DES projects

The comparison between the proposed framework and other documented solutions is summarized in **Table 1**. The proposed framework is represented by the column labeled DESI because both versions of DESI instantiate the proposed framework. In other words, database driven DESI 1.0 and DESI 2.0 were built based on the concepts and ideas presented in the proposed framework. Specifically, both versions rely on the generalization of data needed for DES projects presented in chapter one.

The criteria used for the comparison include: programming language, year of development, whether the tool requires internal data, ability to collect unavailable data but collectable data, ability to perform calculations on raw data, ability to perform distribution fitting, input data management methodology, disciplines where the tool can be used, the flexibility of the tool, real-time histogram creation, real-time distribution fitting, real-time data visualization, real-time goodness-of-fit tests, real-time autocorrelation diagnostics, ease of implementation, ability to handle speed computations and training requirements.

Table 1 Comparison of documented input data management solutions for DES projects

	MDCAP	GDM-TOOL	MDA	PARTS DATA COLLECTION	DESI
Language	Depends on the tool being used	C#.NET	JAVA	VisualStudio.NET, SQL	Matlab, SQL
Year developed	Depends on the tool being used	2012	2010	2004	2014
Requires internal data	NO	YES	YES	YES	NO
Applicable on	Available data, non-available but collectable data, non-collectable data	Available data	Available data	Available data	Non-available but collectable data, available data if it conforms to DESI format requirements in SQL tables or in text files
Automated collection of non-available but collectable data	NO	NO	NO	NO	YES
Calculations on raw data	YES	YES	YES	YES	YES
Distribution fitting	YES	YES	NO	NO	YES
Applications needed	MANY	ONE	ONE	ONE	ONE
Input data management methodology	A,B	C	C	C	Elements (hybrid) from - A - someone collects data; - B - data saved in intermediate MySQL tables or files; - C - data automatically stored and updated; - D - data is ready to be added to models
Area of focus	All	Manufacturing, production process, factories [14]	Manufacturing, materials flow, factories [15]	Manufacturing [29]	All
Usable in multiple DES areas of applications	YES	NO	NO	NO	YES
Real-time automated data fit and histogram visualization	NO	NO	NO	NO	YES
Real-time automated goodness-of-fit tests	NO	NO	NO	NO	YES
Real-time automated autocorrelation diagnostics	NO	NO	NO	NO	YES
Implementation	Depends on the tools used	Complex due to customized configuration stage; trained staff needed [14]	Complex due to the need to implement mappings of different types [15]	Simple, but dependent on other requirements [29]	Simple as it only requires setting parameters in the connections file if using DESI 1.0. DESI 2.0 implementation is even simpler, as it only involves following the installation wizard.
Special training requirements	Depends on the tools used so complexity is variable. Requires knowing more than one tool at a time.	Technical staff required with considerable knowledge of the underlying data and manufacturing system being modeled [14]	Working knowledge of the d3fact simulator and programming skills required in order to understand API, Java script, JDBC, etc. [15]	None beyond knowing where the information resides.	None beyond knowledge about the system being modeled and the model logic.
Average speed calculations	Depends on the tool used	NO	NO	NO	YES, but only on DESI 2.0

Table 1 can also be regarded as a summary of the state-of-the-art in DES input data management automation. DESI has the disadvantage that it is not meant to work with data from internal systems; the reason DESI does not work with internal systems is because there are solutions such as GDM-Tool or MDA that already do that. Within **Table 1** there is a column devoted to MDCAP, as it is the most widely used approach for DES input data management.

DESI not only needs to provide an innovative alternative to fill the gaps of other documented automated solutions but it also needs to provide a viable alternative to MDCAP. **Table 1** shows the areas where the proposed framework and the DESI interface have advanced the state-of-the-art in automated input data management for DES projects.

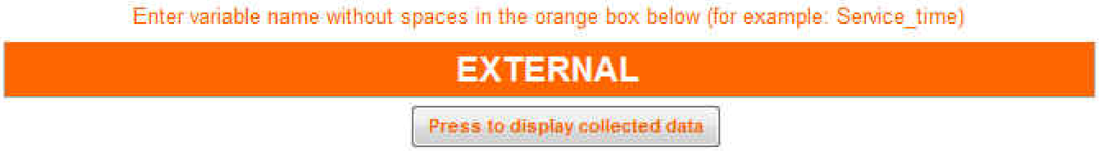
A key contribution of the proposed framework and of the DESI interface is the fact that input data management automation can be achieved in DES applications beyond manufacturing. DESI can be used to automate data collection and processing for modeling non-manufacturing systems such as emergency rooms, airport security checkpoints, lines at a theme park, lines at a grocery store and traffic, just to name a few examples.

The information in **Table 1** shows that DESI advances the state-of-the-art in input data management solutions for DES projects because it covers areas that other tools do not cover. Because the programming of both versions of DESI is based on the proposed integrated framework, it can be affirmed that the proposed integrated framework presented in this research investigation is a significant contribution to the field of modeling and simulation.

Comparison to Standalone Probability Fitting Software

Even though the focus of this research investigation is on solutions that automate all or most of the data preparation steps for DES projects within a single tool, the DESI interface can be directly comparable to standalone commercial probability distribution software such as Arena's Input Analyzer by Rockwell Automation [4], Stat::Fit® by Geer Mountain Software [18], EasyFit® by MathWave Technologies or ExpertFit by Averill M. Law and Associates [16]. This means that data not collected using DESI can be fitted, tested and analyzed using the different features offered by DESI such as goodness-of-fit tests, multimodal fitting and autocorrelation diagnostics. The only requirement is that the external data has the right format.

The way to fit external data not collected using DESI 2.0 is by saving the data as a text file in the folder `C:\DESI\DATA` if working on a pc or in the folder `/Applications/DESI/DATA` if working on a mac. The text file should include one continuous data value per line and should be named `VN_service_times.txt`, where VN stands for any variable name selected by the user. Ideally the variable name should not be shared with previously collected data in order to avoid any confusion, but that will not prevent DESI 2.0 from fitting the external data. Finally, press the button labeled `Press to display collected data` in order to fit the external variable. **Figure 27** shows the orange section of DESI 2.0 and the fitting the hypothetical variable not collected using DESI called EXTERNAL.



Enter variable name without spaces in the orange box below (for example: Service_time)

EXTERNAL

Figure 27 Use DESI to fit external data not collected by DESI

DESI 1.0 can also fit data not collected by DESI 1.0, but the continuous data need to be saved in a MySQL table instead of saving it in a text file. Once the table has been created, it needs to be named following the naming convention `DESI_ST_VN.ST_VN`, as detailed in chapter four. For the example displayed in **Figure 27**, the table in MySQL would be named `DESI_ST_EXTERNAL.ST_EXTERNAL.`

Summary of Comparisons

This chapter presented details about documented solutions for DES input data management automation and how DESI, programed following the proposed framework, compares with them. Documented tools for input data management automation other than DESI focus on manufacturing applications of DES and because they rely on internal data, their applicability outside of manufacturing is limited. These solutions require extensive implementation efforts and detailed customization. Documented solutions prior to DESI are methodology (C) type of solutions. Most DES practitioners acknowledge their desire for the potential automation capabilities offered by methodology (C) solutions. Nevertheless the difficulties associated with these solutions have prevented their proliferation to the extent that 80% of surveyed practitioners still rely on manual approaches for input data management, e.g. methodology (A) and (B) solutions.   DESI was also compared to standalone probability distribution fitting tools. These sophisticated standalone tools often require data pre-processing in order to fit theoretical distributions to the data.

The DESI interface was presented as an innovative option that addresses shortcomings of previously documented solutions. The originality of DESI opens the window for automation beyond manufacturing, advancing the state-of-the-art in DES input data processing.

CHAPTER SIX: EVALUATION OF THE PROPOSED FRAMEWORK

Evaluation Criteria

This chapter presents discrete event simulation projects of reasonable complexity completed using the proposed framework for DES input data management. The goal is to measure the impact of using DESI in the time devoted to input data management. The simulations for each project were carried out using Arena discrete event simulation software. The projects do not include 3D representations of the modeled system; in other words, a project is completed after the model logic has been implemented in the Arena simulation models and the models accurately reflect the real world system. The relevant statistics from each project are grouped in **Table 2**. All projects use significance level of 0.05.

Case Studies

This section presents projects completed using DESI and Arena discrete event simulation software. The data were gathered and fitted using the DESI interface. The relevant metric from each project is the percentage of each project's time devoted to input data management.

Each projects' time-to-completion has been distributed among the following general activities: input data management, model logic development, Arena model creation and transportation to the location of the modeled system. The literature indicates that on average, 31% of DES projects' time is devoted to input data management [5, 33]. The null hypothesis of this analysis is that the proposed framework produces a mean percentage statistically equal to

31% versus the alternative hypothesis that the calculated mean percentage is statistically smaller than 31%. The methodology for the experiment consists of:

1) Measuring the percentage time devoted to input data management for each of the projects presented in this chapter;
2) Calculating the sample mean of the percent time devoted to input data management;
3) Testing the null hypothesis using a one-sided t-test about the mean since the population variance of the percentage devoted to DES input data management is unknown. The one-sided t-test relies on the reasonable assumption that the population of percentages devoted to input data management is normally distributed [43].

Rejecting the null hypothesis in favor of the alternative implies a significant positive impact from using the proposed framework. Formally the test can be defined as follows,

$$H_0 : \mu = .31 \; vs \; H_1 : \mu < .31 \tag{6.1}$$

Based on the results from **Table 2**, the average percentage of time devoted to input data management using DESI is 21%. The calculation of the t-statistic takes the following form

$$t = \frac{.21 - .31}{\frac{.1061}{\sqrt{11}}} \tag{6.2}$$

The t-statistic from (6.2) is -3.13 and produced a p-value of 0.005, which is smaller than the level of significance 0.05, hence H_0 is rejected in favor of H_1. Test details are summarized in **Table 3**.

As presented in chapter three, when using DESI, the time delays arise from the unavoidable wait for the events to happen η_{ρ} and not because of post-collection data processing represented by Λ_{ρ}. All the projects summarized in **Table 2** are presented in detail in subsequent sections of this chapter.

Table 2 Time allocation to complete the evaluation DES projects

PROJECT	INPUT DATA MANAGEMENT	MODEL LOGIC DEVELOPMENT	ARENA MODEL CREATION	TRANSPORT TO LOCATION	TOTAL HOURS	PERCENT USED FOR INPUT DATA MANAGEMENT
MANTA	4	3	21	2	30	13%
WAWA	2	1	4	1	8	25%
STARBUCKS DRIVE THRU	.5	.25	1	.5	2.25	22%
ALTAMONTE MALL FOOD COURT	2	1	10	.5	13.5	15%
SANFORD AIRPORT	.5	.5	1.5	.5	3	17%
PUBLIX CASHIERS	1.5	.5	2.5	.5	5	30%
STARBUCKS COFFEE SHOP	1.5	.5	2.5	.5	5	30%
CHASE BANK DRIVE THRU	1.5	.5	1	.5	3.5	43%
ORLANDO INTL AIRPRT	.5	1	4.5	2	8	6%
COPAAIRLINES	1.5	.5	2.5	2	7.5	20%
CHIPOTLE	.5	1	3	.5	5	10%

Table 3 t-test results

t-statistic	Sample mean	Standard deviation	Degrees of freedom	Critical value	p-value
-3.13	.21	0.1061	10	-1.8125	0.0053

Seaworld Orlando Manta Roller Coaster

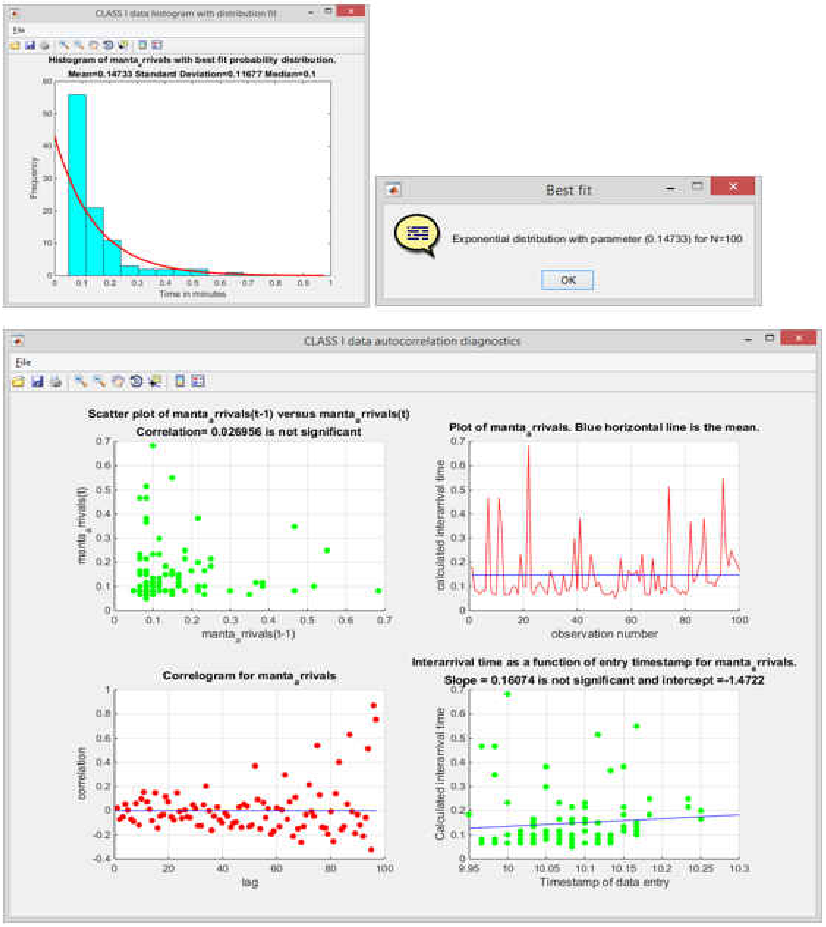
This project was carried out at Seaworld Orlando and models the queue forming process of Manta roller coaster. This project shows the positive impact on queues of having two loading stations instead of one. In order to complete this project, the necessary data were:

- Data on IID arrivals collected using DESI's blue section, which produced the results displayed in **Figure 28**.

- Distribution of number of entities per arrival collected using DESI's green section as displayed in **Figure 29**. The information in **Figure 29** reads as follows: 8.33% of the time there were 4 persons arriving at the roller coaster, 41.7% of the time there were 2 persons arriving at the roller coaster, and so on.
- Data on IID Manta car load times measured using the orange section of DESI and displayed in **Figure 30**. Because Arena does not support all the distributions available in DESI, for convenience only the Lognormal distribution was selected in the left panel of DESI.
- Data on IID quick queue arrivals using the blue section of DESI presented in **Figure 31**.

The full Manta ride takes 2 minutes to complete and this is part of the model logic. Entities are dynamically assigned to each Manta car in batches of 32 riders. This project assumes two Manta cars because that was the situation the day of the data gathering. Both cars were observed to be working at full capacity, and each car can carry as many as 32 riders. **Figure 32** shows the Manta model in Arena DES software.

The Arena model features two entity creation modules in order to define priority attributes based on regular and "quick queue" arrivals. The model logic loads each of the two Manta loading stations based on car availability and queue sizes. The cars share the same steel track and the busy car seizes the track once the ride starts. The Arena model displays the busy steel track utilization with the green square shown in **Figure 32**, which turns white when the steel track is idle. For this project, 13% of the entire project's time was devoted to input data management and the total duration of this project was approximately 30 hours. More information on this project is presented in **Table 2**.



Histogram of manta_arrivals with best fit probability distribution.
Mean=0.14733 Standard Deviation=0.11677 Median=0.1

Exponential distribution with parameter (0.14733) for N=100

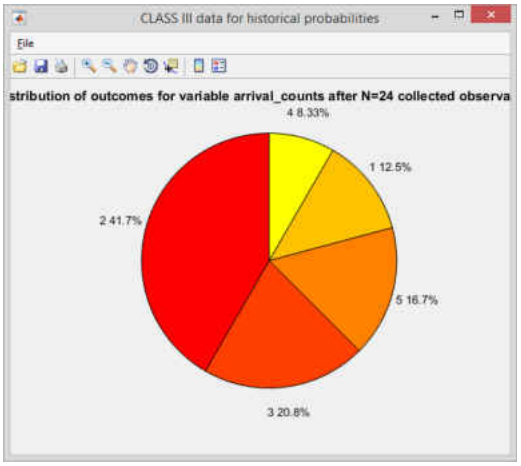
Scatter plot of manta_arrivals(t-1) versus manta_arrivals(t)
Correlation= 0.026956 is not significant

Plot of manta_arrivals. Blue horizontal line is the mean.

Correlogram for manta_arrivals

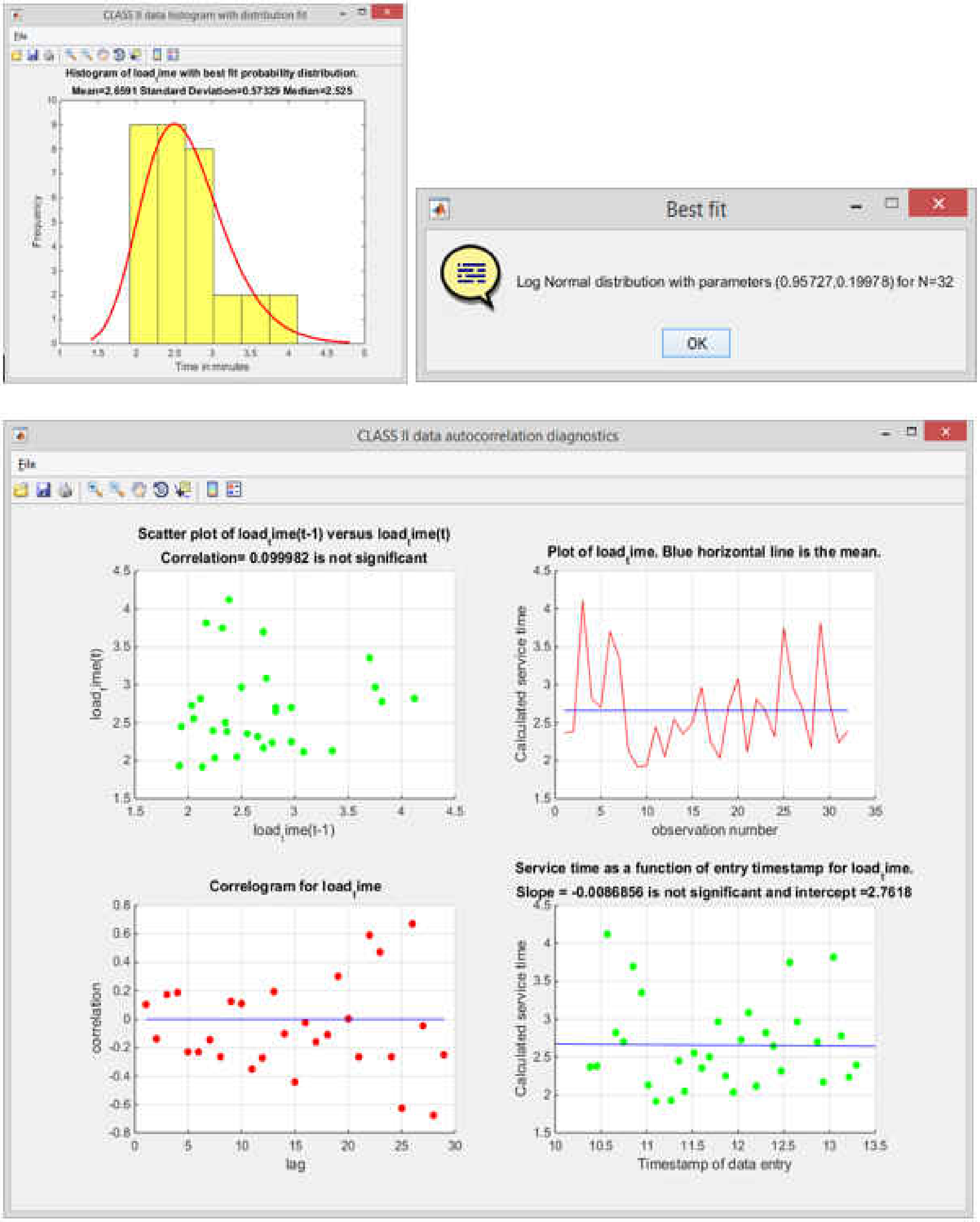
Interarrival time as a function of entry timestamp for manta_arrivals.
Slope = 0.16074 is not significant and intercept =-1.4722

Figure 28 IID Manta arrivals and respective autocorrelation diagnostics.



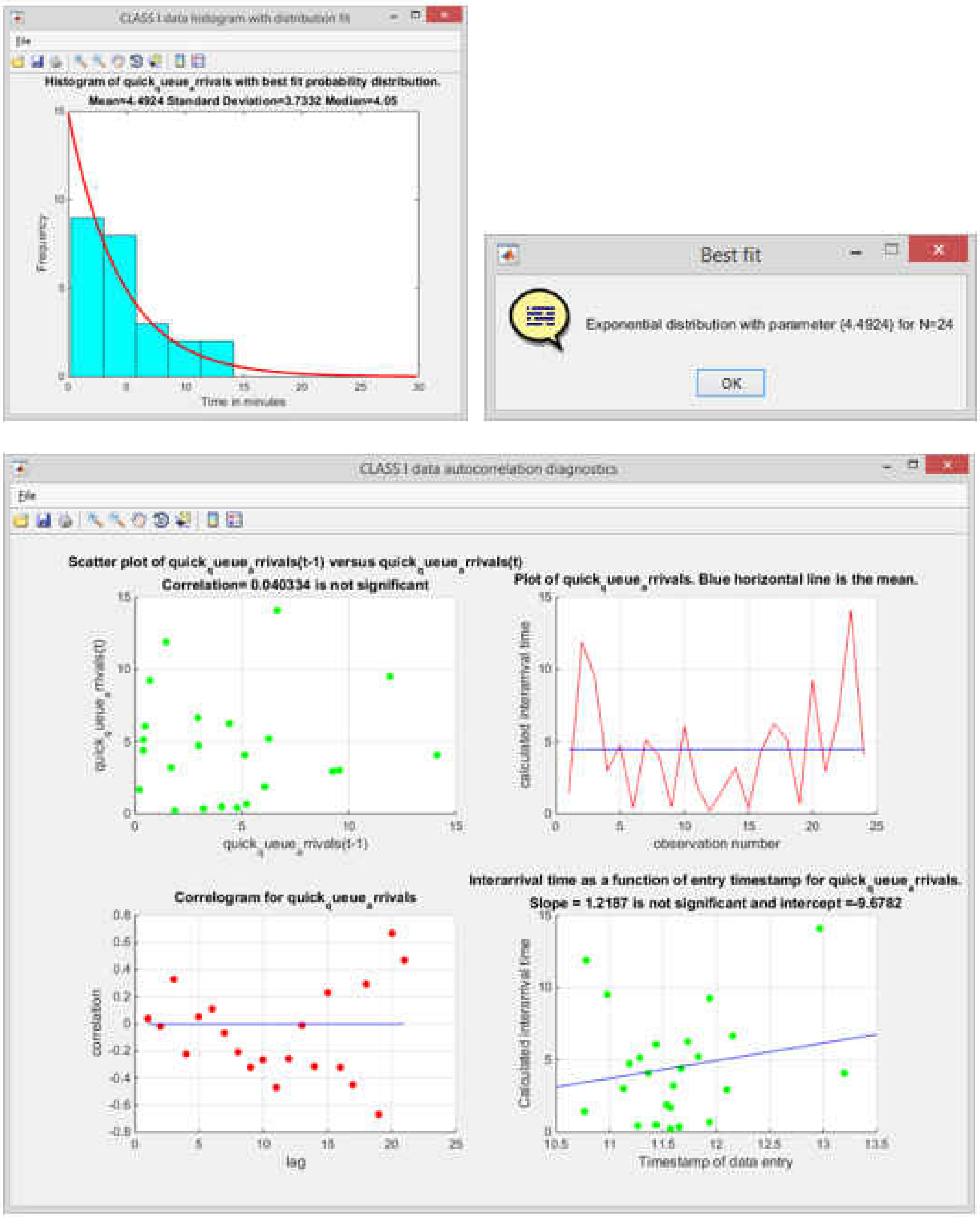
stribution of outcomes for variable arrival_counts after N=24 collected observa

Figure 29 Entity counts per arrival into Manta.


CLASS II data histogram with distribution fit
File
Histogram of load$_t$ime with best fit probability distribution.
Mean=2.6591 Standard Deviation=0.57329 Median=2.525
Frequency
Time in minutes
Best fit
OK
CLASS II data autocorrelation diagnostics
File
Scatter plot of load$_t$ime(t-1) versus load$_t$ime(t)
Correlation= 0.099982 is not significant
load$_t$ime(t)
load$_t$ime(t-1)
Plot of load$_t$ime. Blue horizontal line is the mean.
Calculated service time
observation number
Correlogram for load$_t$ime
correlation
lag
Service time as a function of entry timestamp for load$_t$ime.
Slope = -0.0086856 is not significant and intercept =2.7618
Calculated service time
Timestamp of data entry

Log Normal distribution with parameters (0.95727,0.19978) for N=32

Figure 30 IID Manta car loading times and respective autocorrelation diagnostics.


CLASS I data histogram with distribution fit
File
Histogram of quick$_q$ueue$_a$rrivals with best fit probability distribution.
Mean=4.4924 Standard Deviation=3.7332 Median=4.05
Frequency
0
5
10
15
0
5
10
15
20
25
30
Time in minutes
Best fit

Exponential distribution with parameter (4.4924) for N=24

OK
CLASS I data autocorrelation diagnostics
File
Scatter plot of quick$_q$ueue$_a$rrivals(t-1) versus quick$_q$ueue$_a$rrivals(t)
Correlation= 0.040334 is not significant
quick$_q$ueue$_a$rrivals(t)
0
5
10
15
quick$_q$ueue$_a$rrivals(t-1)
Plot of quick$_q$ueue$_a$rrivals. Blue horizontal line is the mean.
Calculated interarrival time
0
5
10
15
20
25
observation number
Correlogram for quick$_q$ueue$_a$rrivals
correlation
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
lag
Interarrival time as a function of entry timestamp for quick$_q$ueue$_a$rrivals.
Slope = 1.2187 is not significant and intercept =-9.6782
Calculated interarrival time
10.5
11
11.5
12
12.5
13
13.5
Timestamp of data entry

Figure 31 IID Manta quick queue arrivals and respective autocorrelation diagnostics.

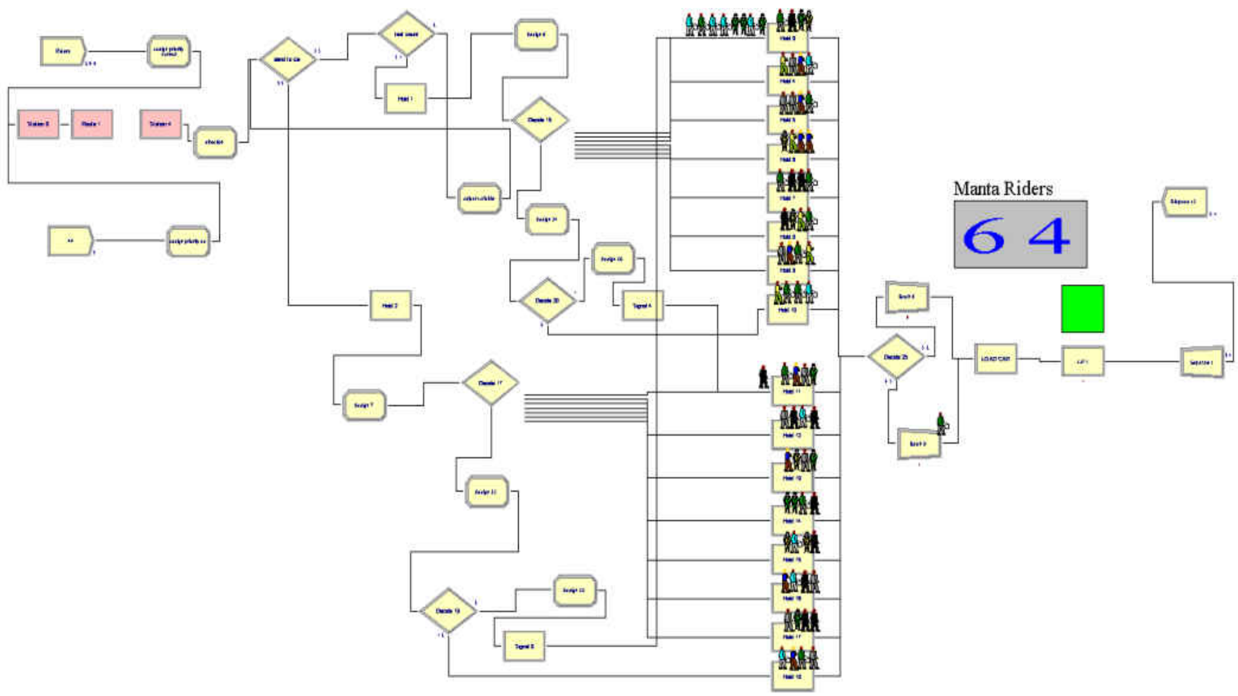


Figure 32 Manta Arena model.

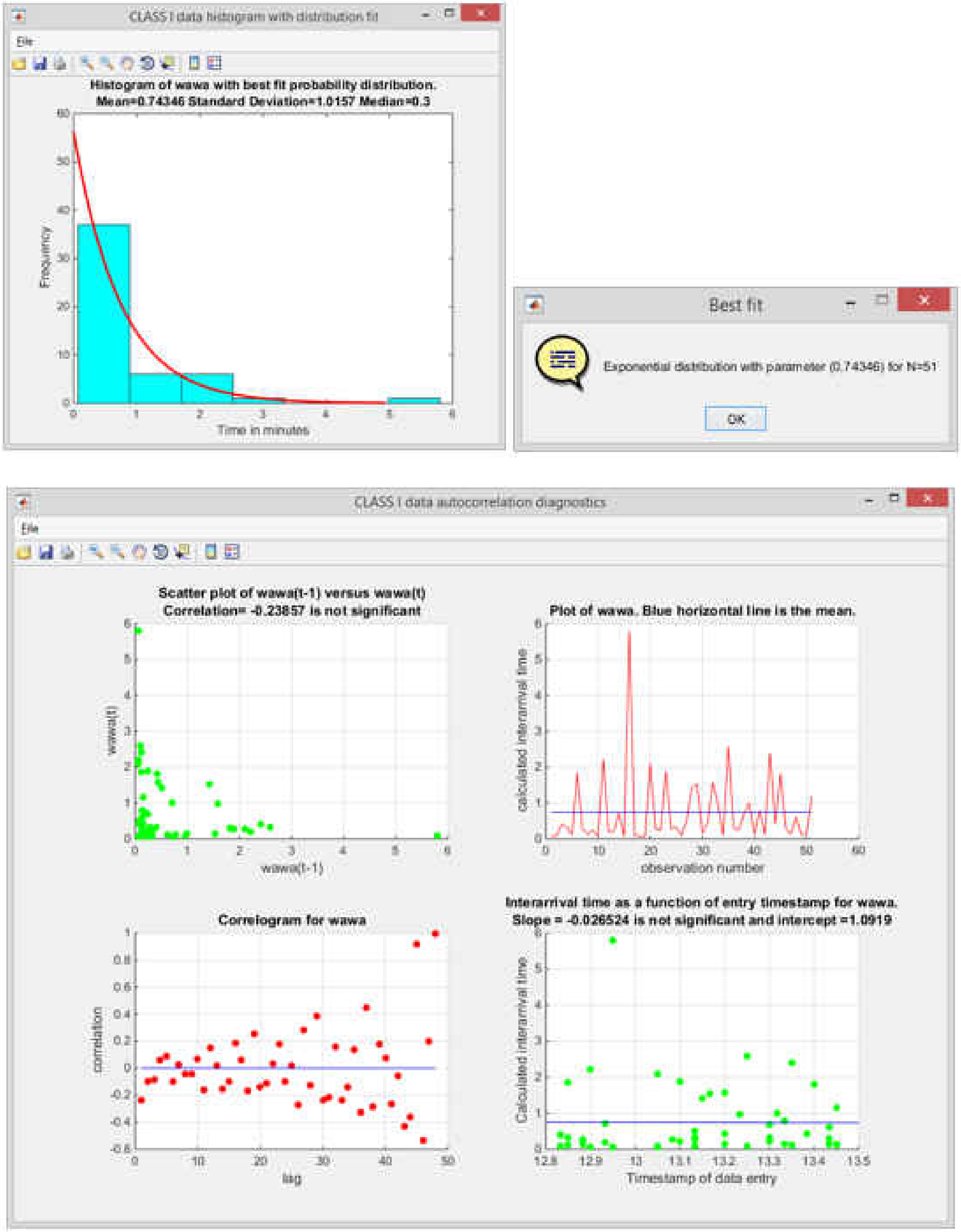
Wawa Gasoline Station

This project was carried out at the Wawa gas station located at 901 N Orlando Ave, Winter Park, FL and seeks to model resource utilization (gas pumps) based on demand. In order to complete this project, the necessary data were:

- Data on IID vehicle arrivals into Wawa from Orlando Avenue gathered using the blue section of DESI as presented in **Figure 33**.
- Data on IID vehicle arrivals into Wawa from Lee Road using the blue section of DESI as presented in **Figure 34**.
- Data on IID gas pumping times gathered using the orange section of DESI as presented in **Figure 35**.

- Data on the percentage of vehicles that stop for gasoline versus the percentage of cars that only arrive at the convenience store gathered using the green section of DESI as presented in **Figure 36**.

The Wawa gas station Arena simulation model is presented in **Figure 37**. The model features 16 resources (gas pumps) and the focus of the analysis is resource utilization based on demand. Cars seize the gas pumps and the utilization status is presented as green/white squares in **Figure 37**. This project presents entities as cars, which can arrive into Wawa from two entrances. Based on collected data, 70% of the incoming cars go for gasoline and the others park in the convenience store depending on parking availability. The convenience store is represented by the orange rectangle in **Figure 37**. The cars that stop for gas seize the free pumps and then leave the system. Cars that go into the convenience store stay for some time and then leave the system. In this project, 25% of the time was spent in input data management out of approximately 8 hours devoted to the entire project. More information on this project is presented in **Table 2**.



CLASS I data histogram with distribution fit

Histogram of wawa with best fit probability distribution.
Mean=0.74346 Standard Deviation=1.0157 Median=0.3

Best fit

Exponential distribution with parameter (0.74346) for N=51

OK

CLASS I data autocorrelation diagnostics

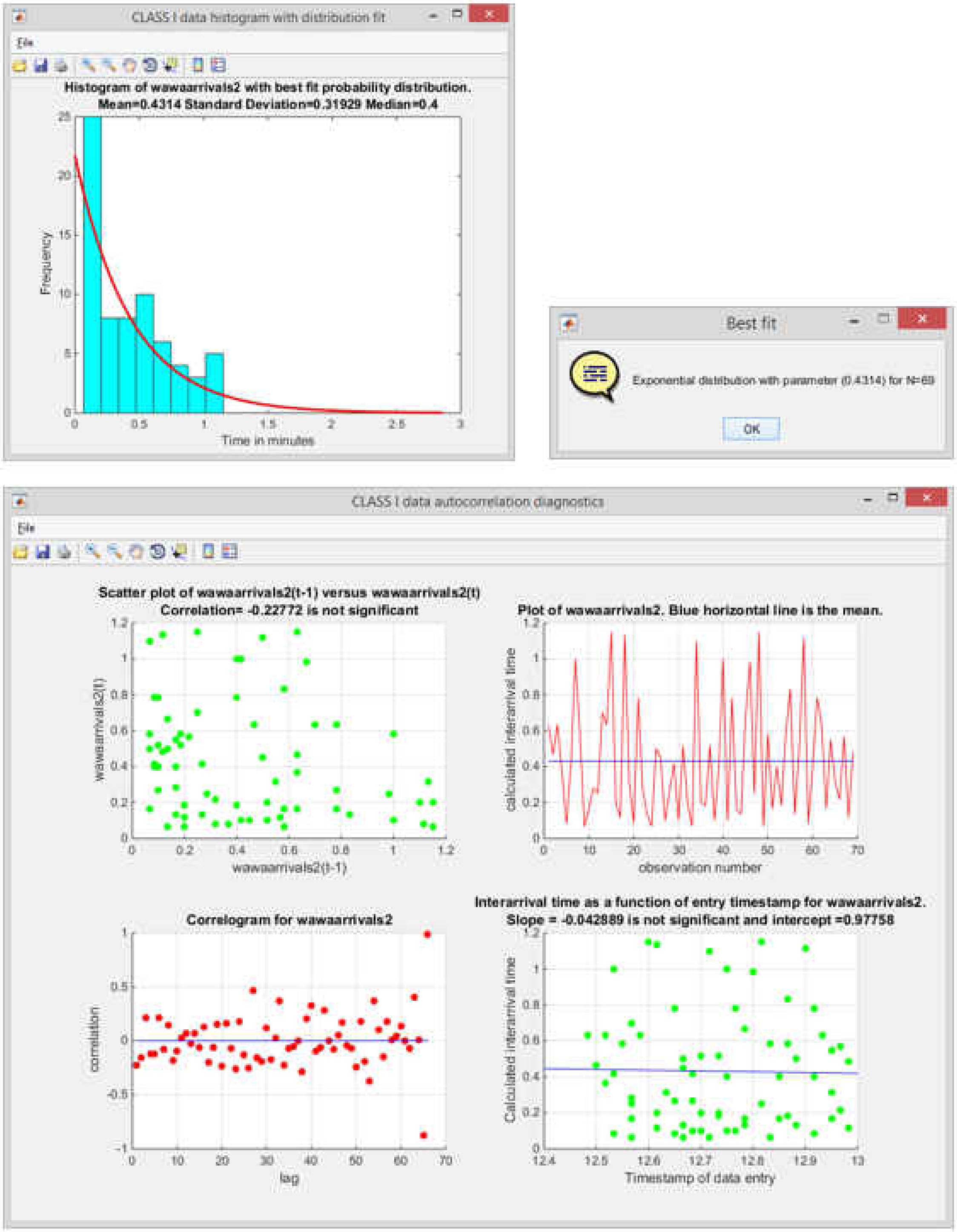
Scatter plot of wawa(t-1) versus wawa(t)
Correlation= -0.23657 is not significant

Plot of wawa. Blue horizontal line is the mean.

Correlogram for wawa

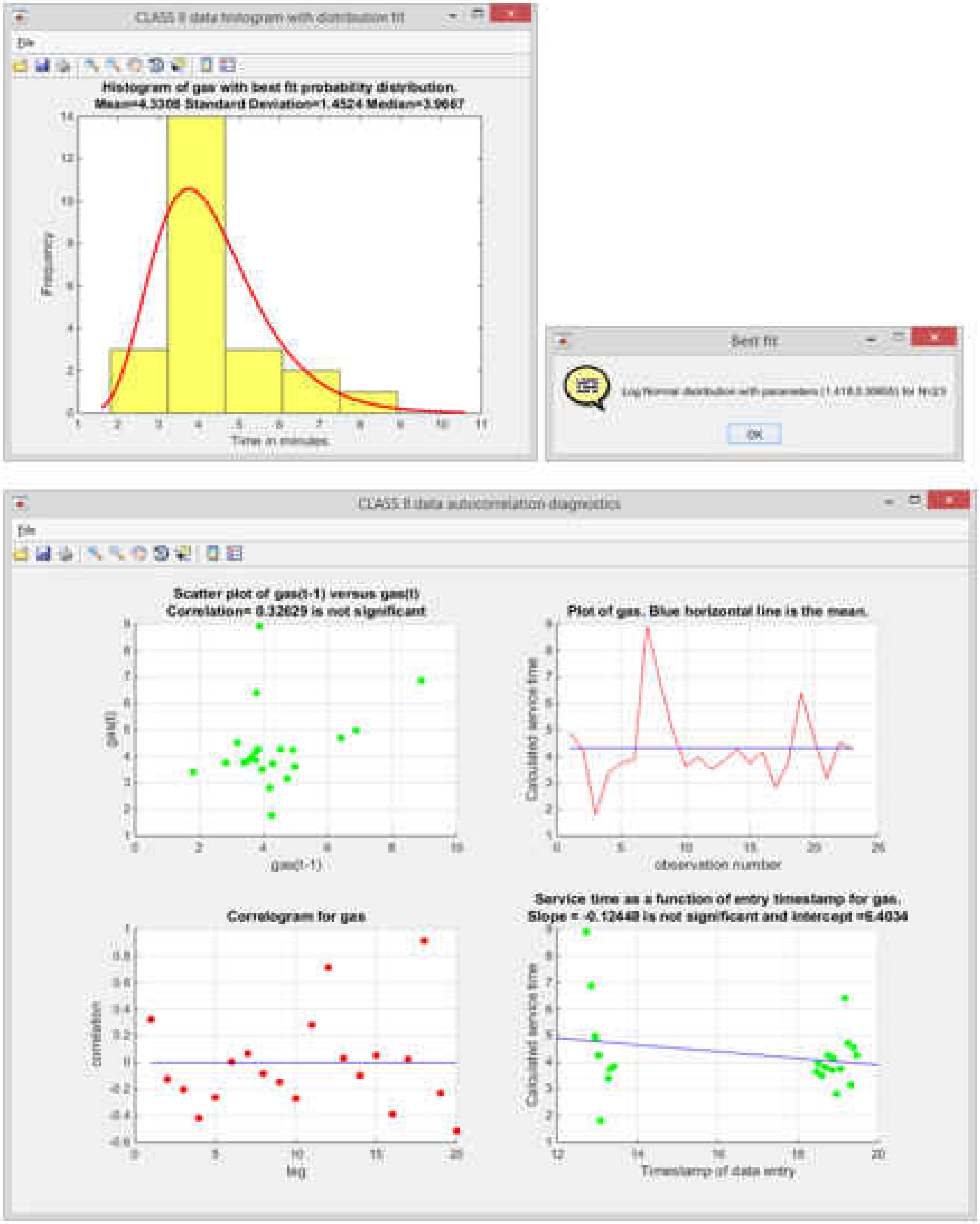
Interarrival time as a function of entry timestamp for wawa.
Slope = -0.026524 is not significant and intercept =1.0919

Figure 33 IID arrivals into Wawa from Orlando Ave and autocorrelation diagnostics.


CLASS I data histogram with distribution fit
File
Histogram of wawaarrivals2 with best fit probability distribution.
Mean=0.4314 Standard Deviation=0.31929 Median=0.4
Frequency
0
5
10
15
20
25
0
0.5
1
1.5
2
2.5
3
Time in minutes
Best fit
OK
CLASS I data autocorrelation diagnostics
File
Scatter plot of wawaarrivals2(t-1) versus wawaarrivals2(t)
Correlation= -0.22772 is not significant
wawaarrivals2(t)
wawaarrivals2(t-1)
0
0.2
0.4
0.6
0.8
1
1.2
Plot of wawaarrivals2. Blue horizontal line is the mean.
calculated interarrival time
observation number
0
10
20
30
40
50
60
70
Correlogram for wawaarrivals2
correlation
lag
-1
-0.5
0
0.5
1
Interarrival time as a function of entry timestamp for wawaarrivals2.
Slope = -0.042889 is not significant and intercept =0.97758
Calculated interarrival time
Timestamp of data entry
12.4
12.5
12.6
12.7
12.8
12.9
13

Exponential distribution with parameter (0.4314) for N=69

Figure 34 IID arrivals into Wawa from Lee Road and autocorrelation diagnostics.



CLASS II data histogram with distribution fit
File
Histogram of gas with best fit probability distribution.
Mean=4.3308 Standard Deviation=1.4524 Median=3.9667
Frequency
0
2
4
6
8
10
12
14
1
2
3
4
5
6
7
8
9
10
11
Time in minutes
Best fit
OK
CLASS II data autocorrelation diagnostics
File
Scatter plot of gas(t-1) versus gas(t)
Correlation= 0.32629 is not significant
gas(t)
gas(t-1)
Plot of gas. Blue horizontal line is the mean.
Calculated service time
observation number
Correlogram for gas
correlation
lag
Service time as a function of entry timestamp for gas.
Slope = -0.12448 is not significant and intercept =6.4034
Calculated service time
Timestamp of data entry

Log Normal distribution with parameters (1.418,0.30858) for N=23

Figure 35 IID gas pumping times and autocorrelation diagnostics.

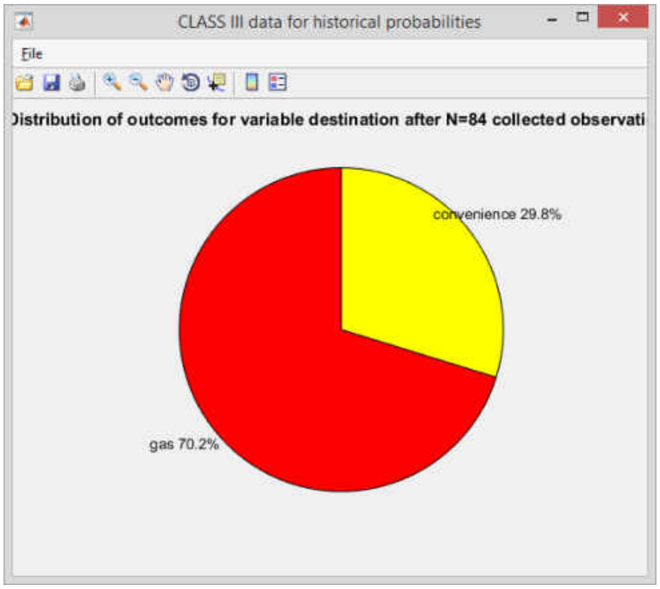

CLASS III data for historical probabilities
File
)istribution of outcomes for variable destination after N=84 collected observati
convenience 29.8%
gas 70.2%

Figure 36 Cars going for gasoline versus cars going to the Wawa convenience store.

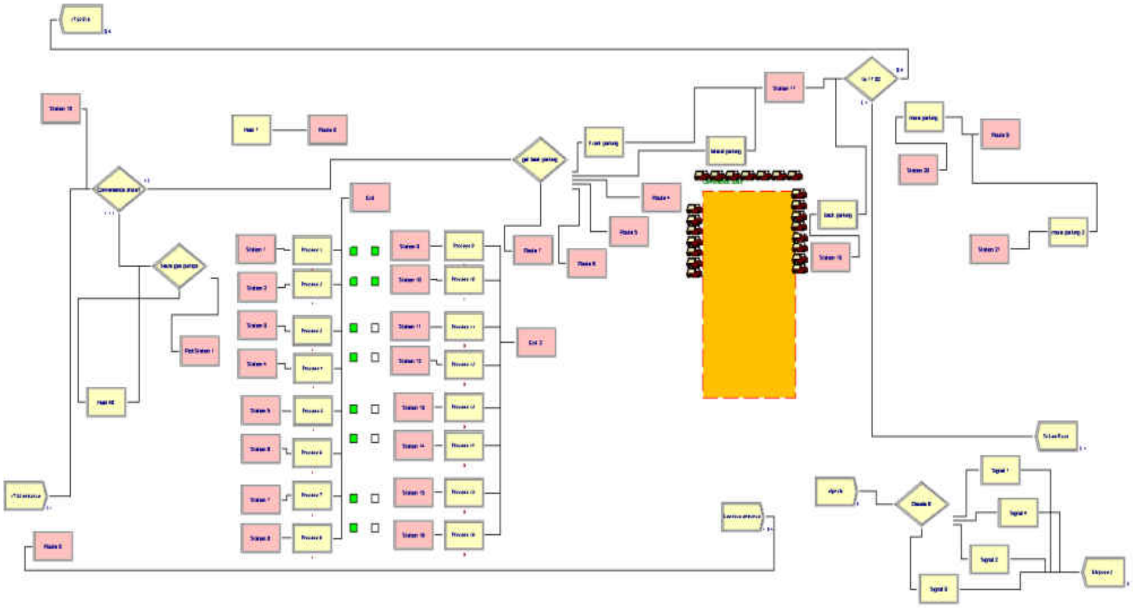


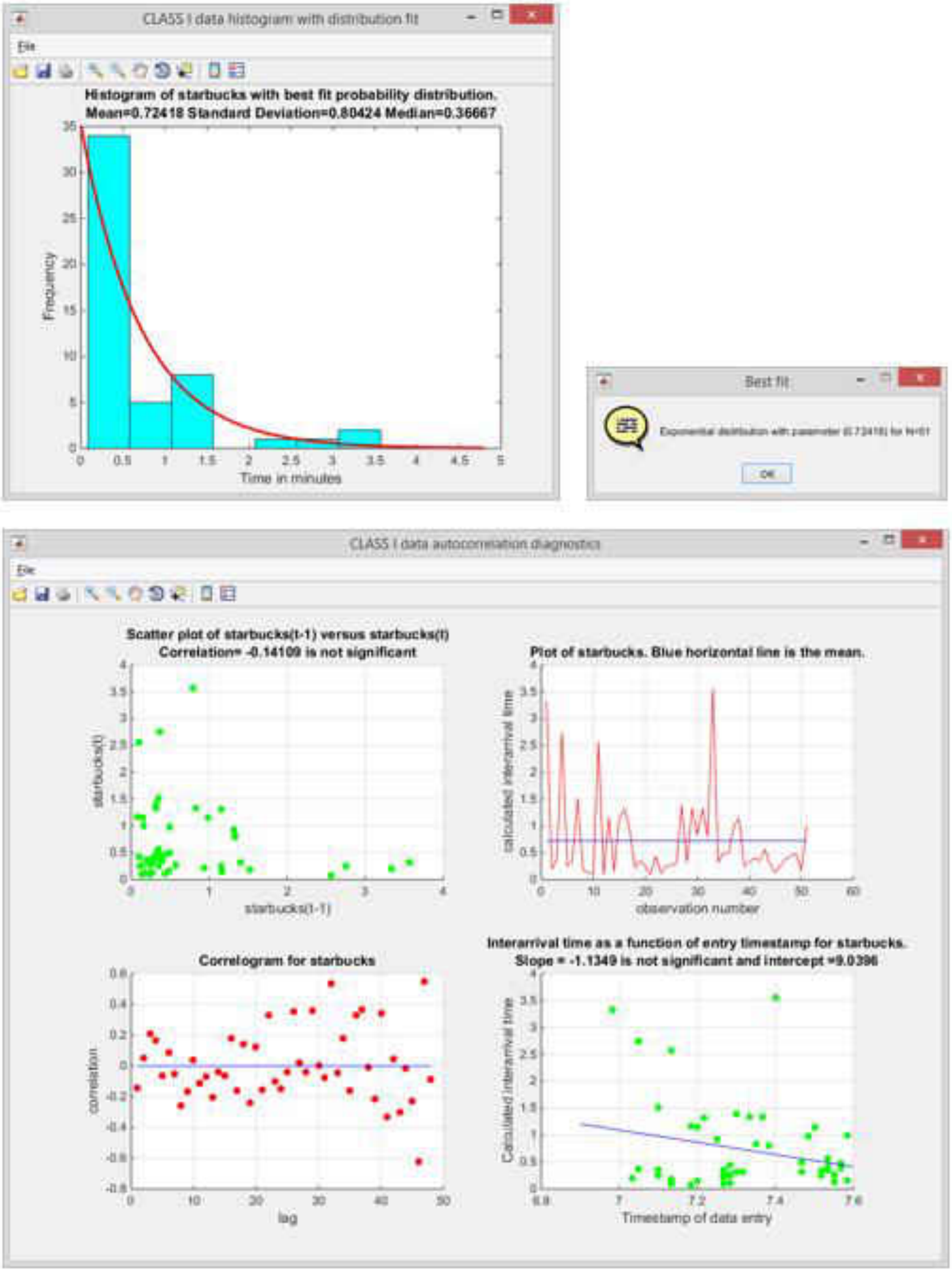
Figure 37 Wawa Arena simulation model.

Winter Park Village Starbucks Coffee Drive Thru

This project was carried out at the Starbucks coffee shop located at 600 N Orlando Ave, Winter Park, FL and seeks to model the drive thru during morning hours. The motivation behind the Starbucks drive thru project is the fact that long car queues impact the traffic into the Winter Park village parking area. In order to complete this project, the necessary data were:

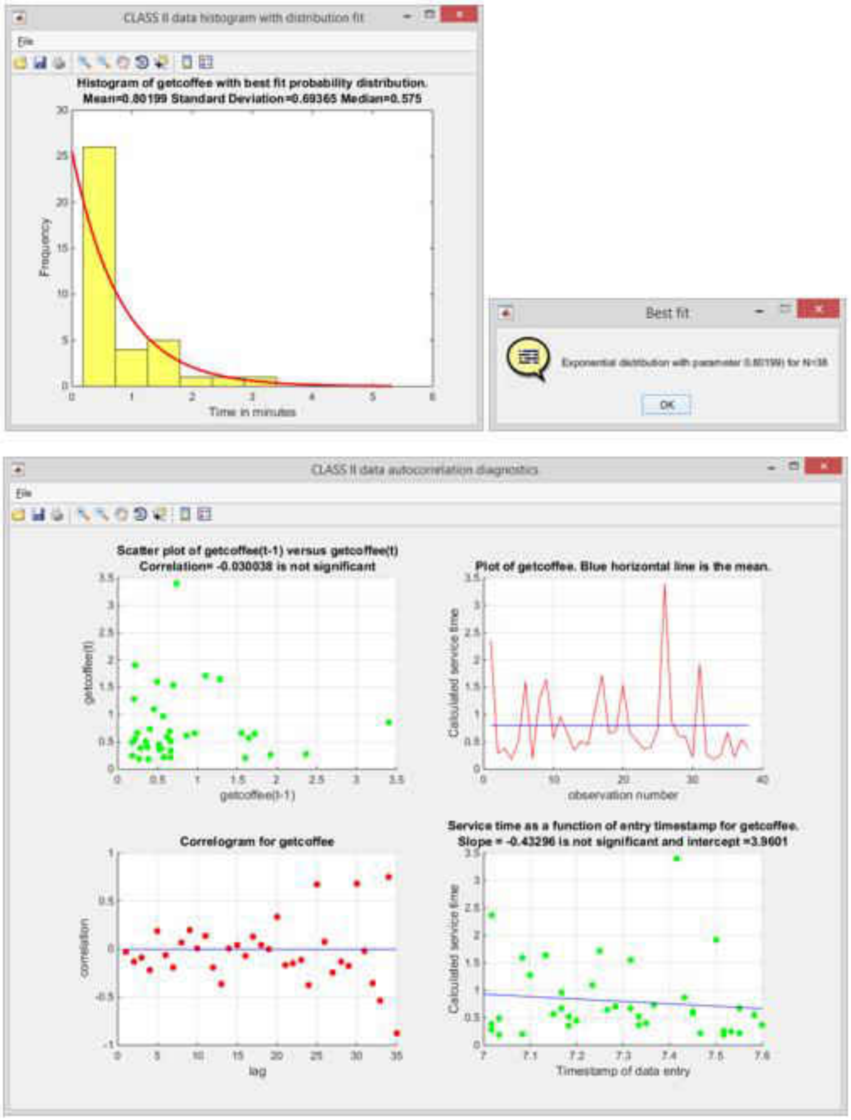
- Data on IID car arrivals to the drive thru collected using the blue section of DESI presented in **Figure 38**.
- Data on queue size in terms of how many cars can fit at a given time between the ordering station and the window where the order is dispatched; current physical queue size is overwhelmed by demand.
- Data on IID wait time to get the order at the window gathered using the orange section of DESI presented in **Figure 39**.

The Starbucks Arena model is displayed in **Figure 40**. The Arena model includes the delay logic when customers spend some time placing their orders; such delay is zero when the order pickup window queue is greater than 4, because customers must wait due to the queue and not because of the normal order delay. The Arena model also displays the busy state of the order pickup window with a green square. The scope of this project focuses in the morning demand of the Starbuck’s drive thru, hence replications only run for three hours. In this project, the percentage time devoted to input data management was 22% out of the entire project’s time of 2 hours and 15 minutes. More information on this project is presented in **Table 2**.


CLASS I data histogram with distribution fit
File
Histogram of starbucks with best fit probability distribution.
Mean=0.72418 Standard Deviation=0.80424 Median=0.36667
Frequency
35
30
25
20
15
10
5
0
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
Time in minutes
Best fit
OK
CLASS I data autocorrelation diagnostics
File
Scatter plot of starbucks(t-1) versus starbucks(t)
Correlation= -0.14109 is not significant
starbucks(t)
starbucks(t-1)
Plot of starbucks. Blue horizontal line is the mean.
calculated interarrival time
observation number
Correlogram for starbucks
correlation
lag
Interarrival time as a function of entry timestamp for starbucks.
Slope = -1.1349 is not significant and intercept =9.0396
Calculated interarrival time
Timestamp of data entry
6.8
7
7.2
7.4
7.6

Exponential distribution with parameter [0.72418] for N=51

Figure 38 IID arrivals into Starbucks drive thru and autocorrelation diagnostics.


CLASS II data histogram with distribution fit
File
Histogram of getcoffee with best fit probability distribution.
Mean=0.80199 Standard Deviation=0.69365 Median=0.575
Frequency
Time in minutes
Best fit
OK
CLASS II data autocorrelation diagnostics
File
Scatter plot of getcoffee(t-1) versus getcoffee(t)
Correlation= -0.030038 is not significant
getcoffee(t)
getcoffee(t-1)
Plot of getcoffee. Blue horizontal line is the mean.
Calculated service time
observation number
Correlogram for getcoffee
correlation
lag
Service time as a function of entry timestamp for getcoffee.
Slope = -0.43296 is not significant and intercept =3.9601
Calculated service time
Timestamp of data entry

Exponential distribution with parameter 0.80199 for N=38

Figure 39 IID Starbucks drive thru service times and autocorrelation diagnostics.

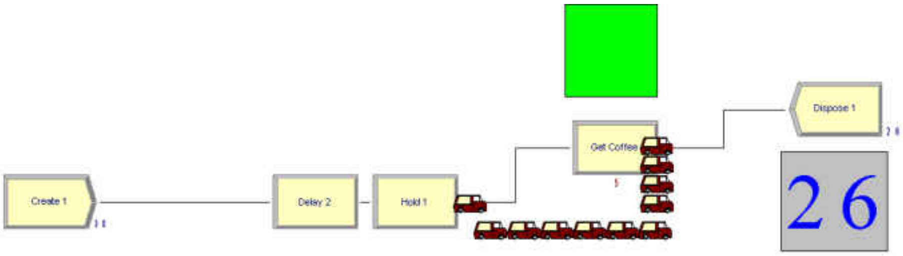


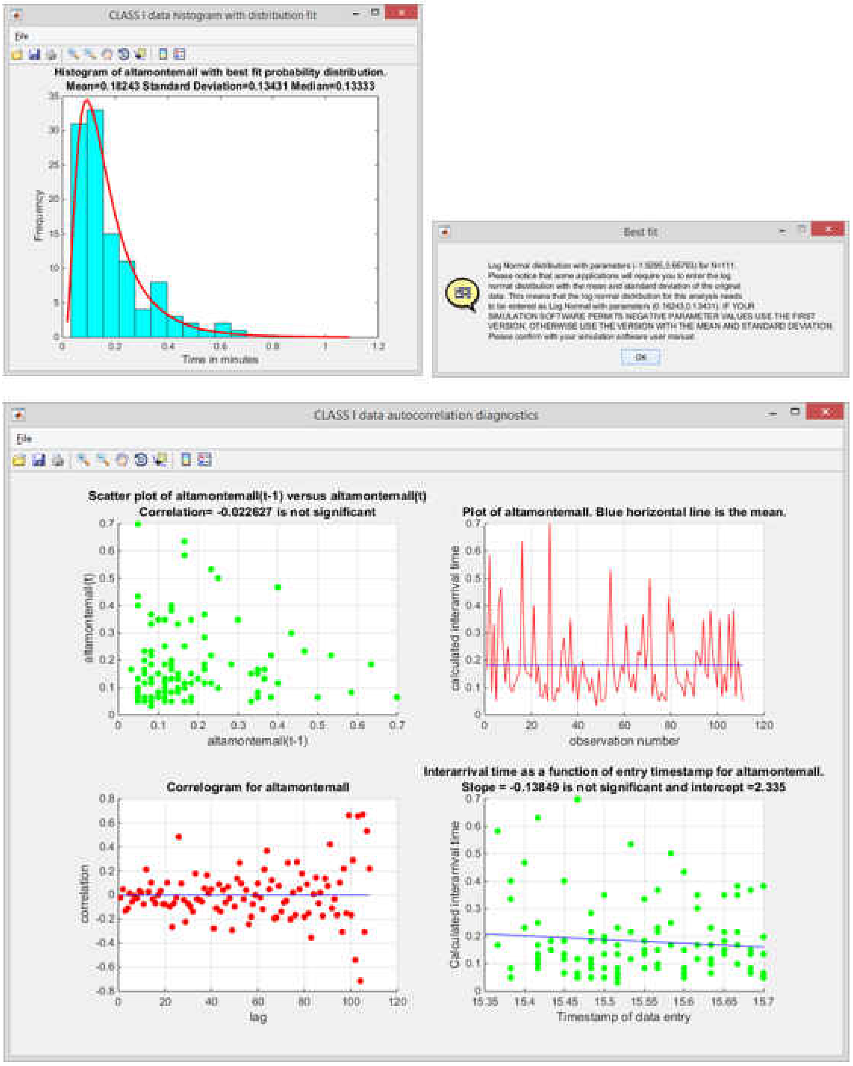
Figure 40 Starbucks Arena model.

Altamonte Mall Food Court

This project was carried out in the food court of the Altamonte Mall, in Altamonte Springs, Florida. This model focuses on the demand for seats to eat in the food court. There are a approximately 400 available seats, distributed among100 available tables. Seats and tables can be moved therefore seats allocation is flexible depending on party sizes. In order to complete this project, the necessary data were:

- Data on IID arrivals to the food court gathered using the blue section of DESI, as presented in **Figure 41**.
- Data on arriving party sizes gathered using the green section of DESI, as presented in **Figure 42**.
- Data on IID food serving times gathered using the orange section of DESI, as presented in **Figure 43**.
- Data on IID payment times gathered using the orange section of DESI, as presented in **Figure 44**.
- Data on number of tables and quantity of seats per table gathered using the green section of DESI, as presented in **Figure 45**.

The Altamonte mall food court Arena model is presented in **Figure 46**. This logic requires the use of 13 sub-models in order to model the 13 food stations. One of the 13 sub-models is shown in **Figure 47**. In the food serving process, individual customers arrive at the different fast food stations such as Taco Bell, Subway, Sbarro, etc. and they are helped by staff plus a cashier who receives the payment. There were several staff members on each station, but only two staff members were observed to help customers at a given time, plus the cashier. Then customers are batched based on party size probabilities and proceed to the tables. The logic of the Arena model assigns party size as an attribute, so when customers seize the dining seats, they seize the right number of seats based on the party size. The seating space resource has a capacity of 400 seats and each party's size reduces the number of available seats by the number of the party size. The model keeps track of the entities entering the dining area and the number of busy and free seats.15% of this project's time was spent in input data management out of a total of 13.5 hours. More details about this project can be found in **Table 2**.



Log Normal distribution with parameters (-1.9295,0.66793) for N=111.
Please notice that some applications will require you to enter the log
normal distribution with the mean and standard deviation of the original
data. This means that the log normal distribution for this analysis needs
to be entered as Log Normal with parameters (0.18243,0.13431). IF YOUR
SIMULATION SOFTWARE PERMITS NEGATIVE PARAMETER VALUES USE THE FIRST
VERSION, OTHERWISE USE THE VERSION WITH THE MEAN AND STANDARD DEVIATION
Please confirm with your simulation software user manual.

CLASS I data histogram with distribution fit
File
Histogram of altamontemall with best fit probability distribution.
Mean=0.18243 Standard Deviation=0.13431 Median=0.13333
Frequency
0
5
10
15
20
25
30
35
0
0.2
0.4
0.6
0.8
1
1.2
Time in minutes
Best fit
OK
CLASS I data autocorrelation diagnostics
File
Scatter plot of altamontemall(t-1) versus altamontemall(t)
Correlation= -0.022627 is not significant
altamontemall(t)
altamontemall(t-1)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Plot of altamontemall. Blue horizontal line is the mean.
calculated interarrival time
observation number
0
20
40
60
80
100
120
Correlogram for altamontemall
correlation
lag
-0.8
-0.6
-0.4
-0.2
0.2
0.4
0.6
0.8
Interarrival time as a function of entry timestamp for altamontemall.
Slope = -0.13849 is not significant and intercept =2.335
Calculated interarrival time
Timestamp of data entry
15.35
15.4
15.45
15.5
15.55
15.6
15.65
15.7

Figure 41IID arrivals into the Altamonte mall food court and autocorrelation diagnostics.

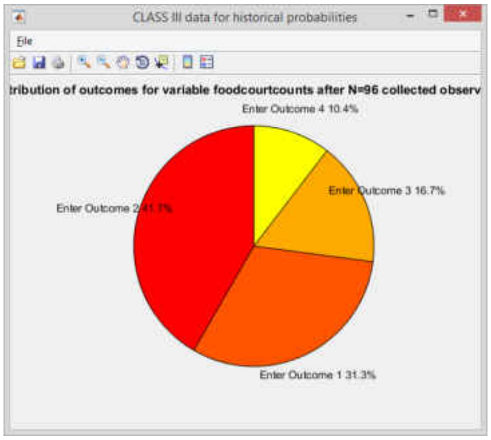
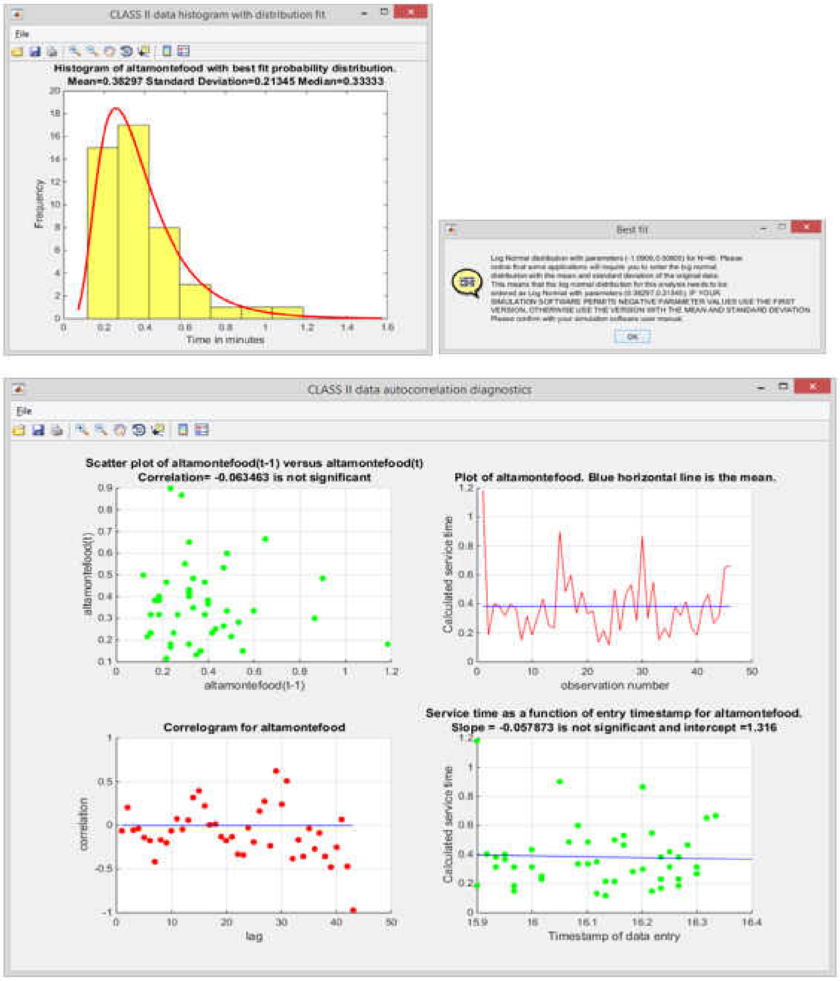
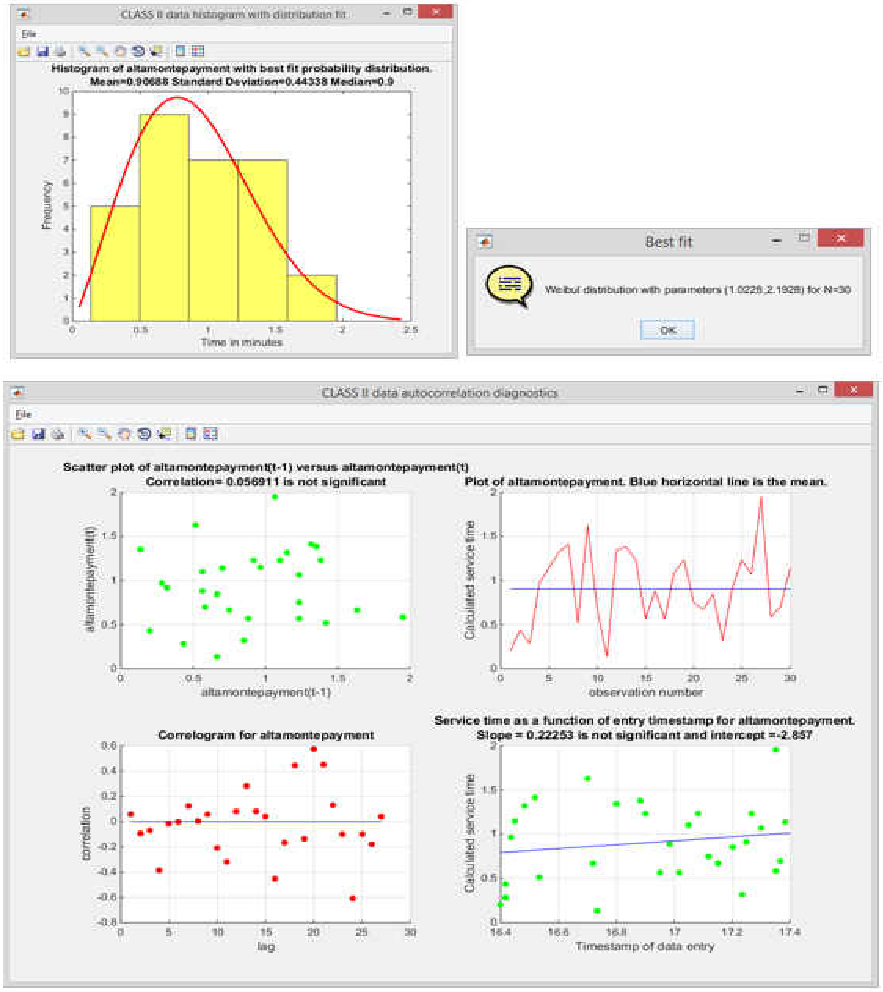

CLASS III data for historical probabilities
File
tribution of outcomes for variable foodcourtcounts after N=96 collected observ
Enter Outcome 4 10.4%
Enter Outcome 3 16.7%
Enter Outcome 2 41.7%
Enter Outcome 1 31.3%

Figure 42 Party size distribution per arrivals into Altamonte mall food court.


CLASS II data histogram with distribution fit
File
Histogram of altamontefood with best fit probability distribution.
Mean=0.38297 Standard Deviation=0.21345 Median=0.33333
Frequency
Time in minutes
0
2
4
6
8
10
12
14
16
18
20
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
Best fit
OK
CLASS II data autocorrelation diagnostics
File
Scatter plot of altamontefood(t-1) versus altamontefood(t)
Correlation= -0.063463 is not significant
altamontefood(t)
altamontefood(t-1)
Plot of altamontefood. Blue horizontal line is the mean.
Calculated service time
observation number
Correlogram for altamontefood
correlation
lag
Service time as a function of entry timestamp for altamontefood.
Slope = -0.057873 is not significant and intercept =1.316
Calculated service time
Timestamp of data entry
15.9
16
16.1
16.2
16.3
16.4

Log Normal distribution with parameters (-1.0906,0.50806) for N=46. Please notice that some applications will require you to enter the log normal distribution with the mean and standard deviation of the original data. This means that the log normal distribution for this analysis needs to be entered as Log Normal with parameters (0.38297,0.21345). IF YOUR SIMULATION SOFTWARE PERMITS NEGATIVE PARAMETER VALUES USE THE FIRST VERSION, OTHERWISE USE THE VERSION WITH THE MEAN AND STANDARD DEVIATION. Please confirm with your simulation software user manual.

Figure 43 IID Service times at fast food stations and autocorrelation diagnostics.


CLASS II data histogram with distribution fit
File
Histogram of altamontepayment with best fit probability distribution.
Mean=0.90688 Standard Deviation=0.44338 Median=0.9
Frequency
Time in minutes
0
0.5
1
1.5
2
2.5
Best fit
OK
CLASS II data autocorrelation diagnostics
File
Scatter plot of altamontepayment(t-1) versus altamontepayment(t)
Correlation= 0.056911 is not significant
altamontepayment(t)
altamontepayment(t-1)
Plot of altamontepayment. Blue horizontal line is the mean.
Calculated service time
observation number
Correlogram for altamontepayment
correlation
lag
Service time as a function of entry timestamp for altamontepayment.
Slope = 0.22253 is not significant and intercept =-2.857
Calculated service time
Timestamp of data entry

Weibul distribution with parameters (1.0228,2.1928) for N=30

Figure 44 IID Payment times at fast food stations and autocorrelation diagnostics.

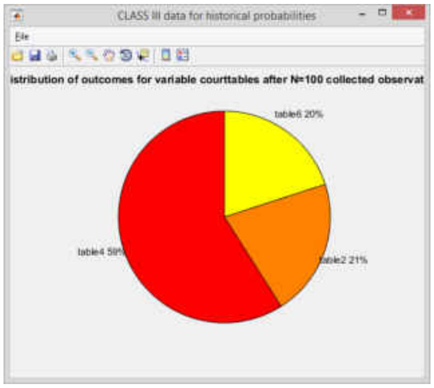


Figure 45 Distribution of seats per table in Altamonte mall food court.

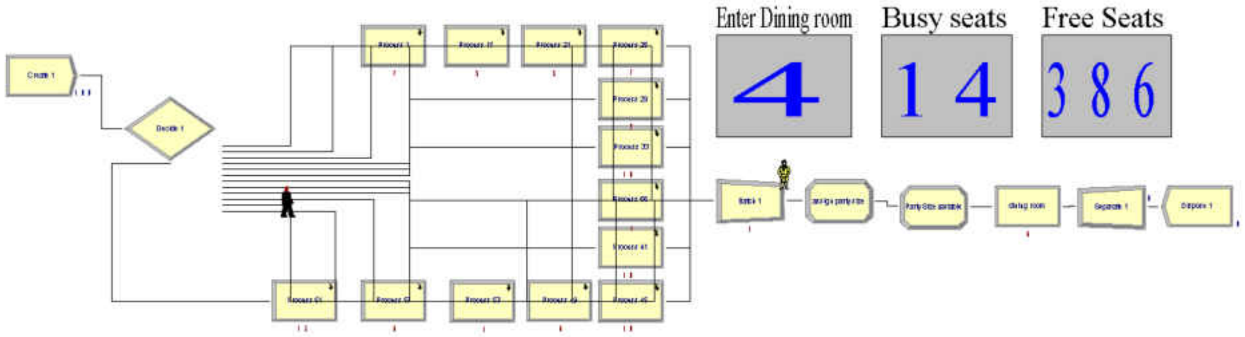


Figure 46 Altamonte food court Arena model.

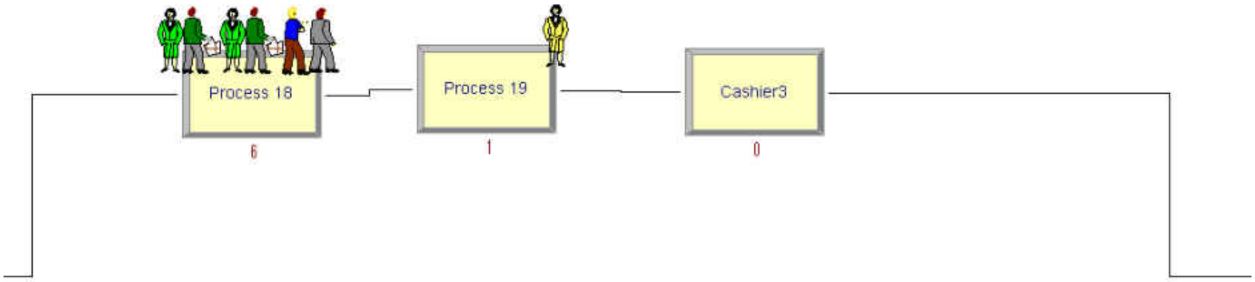
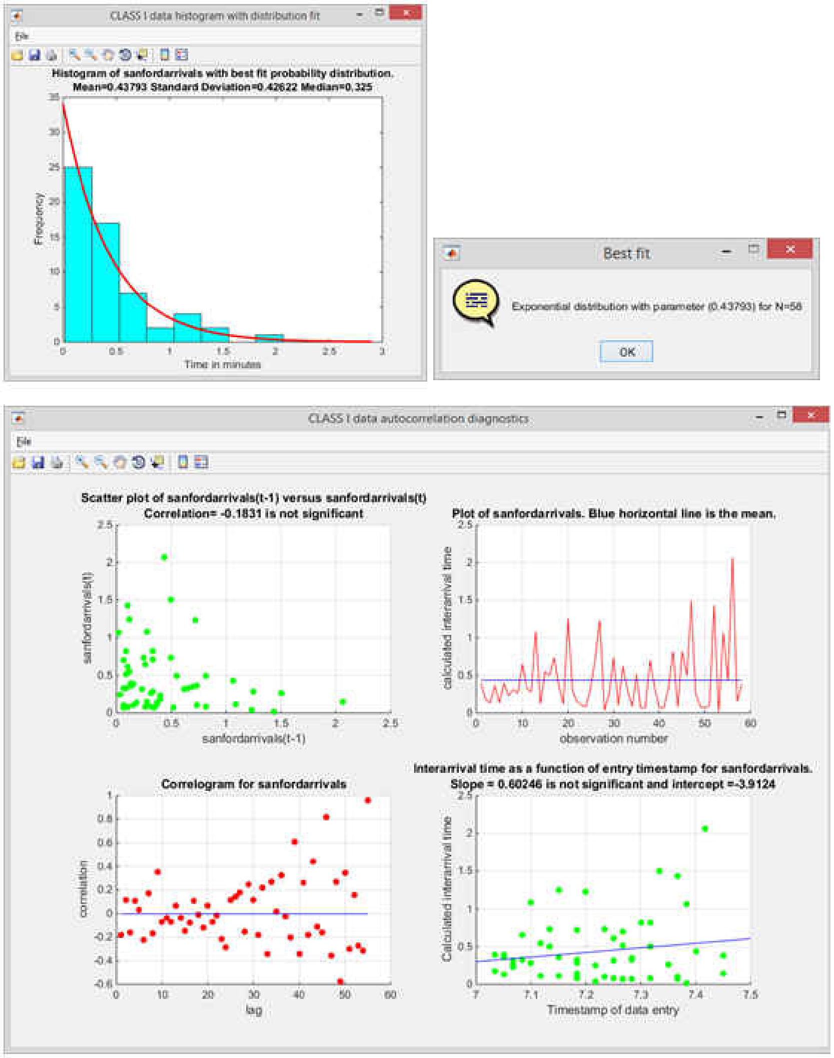


Figure 47 One of the 13 sub-models representing a fast food station in the food court.

Orlando-Sanford International Security Checkpoint

This project was carried out at the Orlando Sanford International Airport, in Sanford, FL. The goal of this project is to simulate the passengers arriving and going thru the security checkpoint of this relatively small airport. At the moment of the data collection, only one of the two available X-ray positions was used to check passengers. Even though the security checkpoint has periods of low demand, the queues can get very large when passengers need to proceed to the gates. As a matter of fact, security lines can get so crowded that passengers are retained in the lower level of the airport if the queue is too crowded. Then passengers are allowed to proceed up the stairs so they can enter the actual security line. In order to complete this project, the necessary data were:

- IID arrivals of ticketed passengers into the security checkpoint presented in **Figure 48**.
- Delay in getting personal items thru the X-ray machine presented in **Figure 49**.



CLASS I data histogram with distribution fit

Histogram of sanfordarrivals with best fit probability distribution.
Mean=0.43793 Standard Deviation=0.42622 Median=0.325

Best fit

Exponential distribution with parameter (0.43793) for N=58

CLASS I data autocorrelation diagnostics

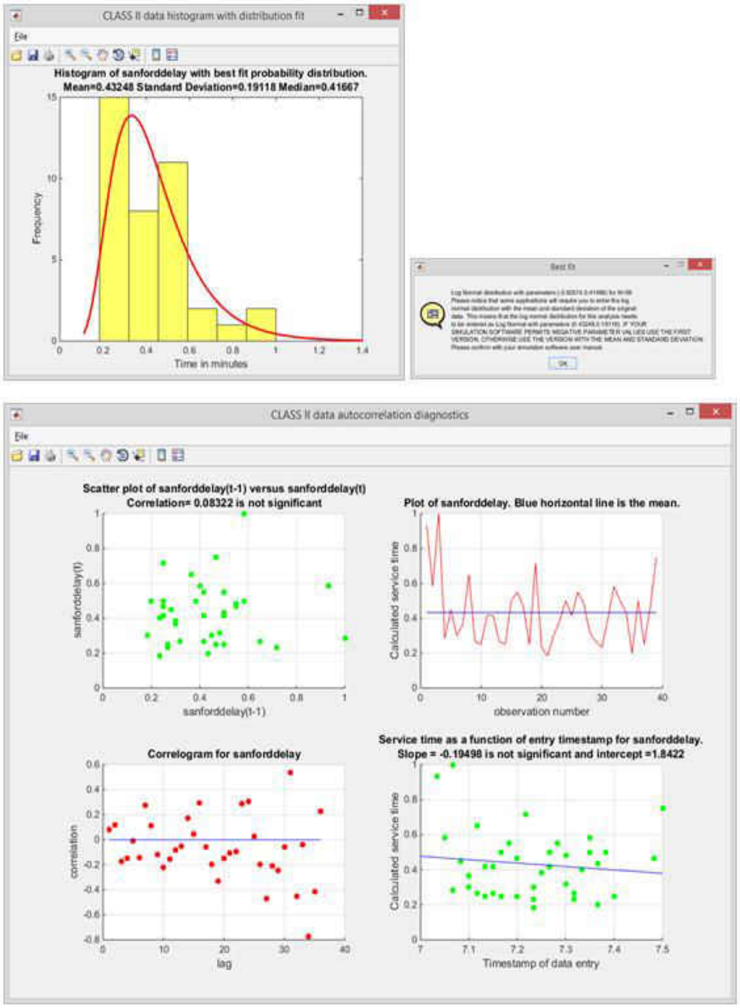
Scatter plot of sanfordarrivals(t-1) versus sanfordarrivals(t)
Correlation= -0.1831 is not significant

Plot of sanfordarrivals. Blue horizontal line is the mean.

Correlogram for sanfordarrivals

Interarrival time as a function of entry timestamp for sanfordarrivals.
Slope = 0.60246 is not significant and intercept =-3.9124

Figure 48 IID Arrivals into Sanford Airport security checkpoint and diagnostics.


CLASS II data histogram with distribution fit
File
Histogram of sanforddelay with best fit probability distribution.
Mean=0.43248 Standard Deviation=0.19118 Median=0.41667
Frequency
Time in minutes
0
5
10
15
0.2
0.4
0.6
0.8
1
1.2
1.4
Best fit
CLASS II data autocorrelation diagnostics
Scatter plot of sanforddelay(t-1) versus sanforddelay(t)
Correlation= 0.08322 is not significant
sanforddelay(t)
sanforddelay(t-1)
Plot of sanforddelay. Blue horizontal line is the mean.
Calculated service time
observation number
Correlogram for sanforddelay
correlation
lag
Service time as a function of entry timestamp for sanforddelay.
Slope = -0.19498 is not significant and intercept =1.8422
Calculated service time
Timestamp of data entry
7
7.1
7.2
7.3
7.4
7.5

Log Normal distribution with parameters (-0.92074,0.41986) for N=38
Please notice that some applications will require you to enter the log
normal distribution with the mean and standard deviation of the original
data. This means that the log normal distribution for this analysis needs
to be entered as Log Normal with parameters (0.43248,0.19118). IF YOUR
SIMULATION SOFTWARE PERMITS NEGATIVE PARAMETER VALUES USE THE FIRST
VERSION. OTHERWISE USE THE VERSION WITH THE MEAN AND STANDARD DEVIATION
Please confirm with your simulation software user manual.

Figure 49 IID delays before going thru the X-ray machine in Sanford International airport.

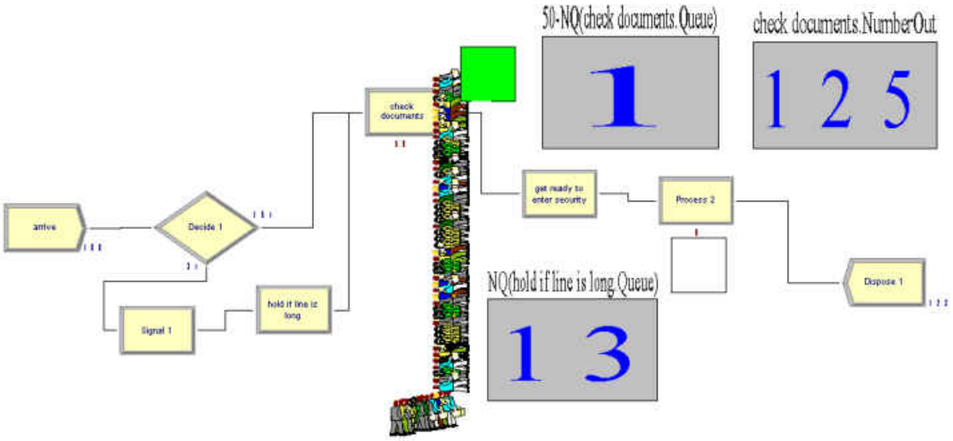

50-NQ(check documents.Queue)
1
check documents.NumberOut
1 2 5
check documents
arrive
Decide 1
Signal 1
hold if line is long
get ready to enter security
Process 2
Dispose 1
NQ(hold if line is long.Queue)
1 3

Figure 50 Arena Model for Sanford airport security checkpoint.

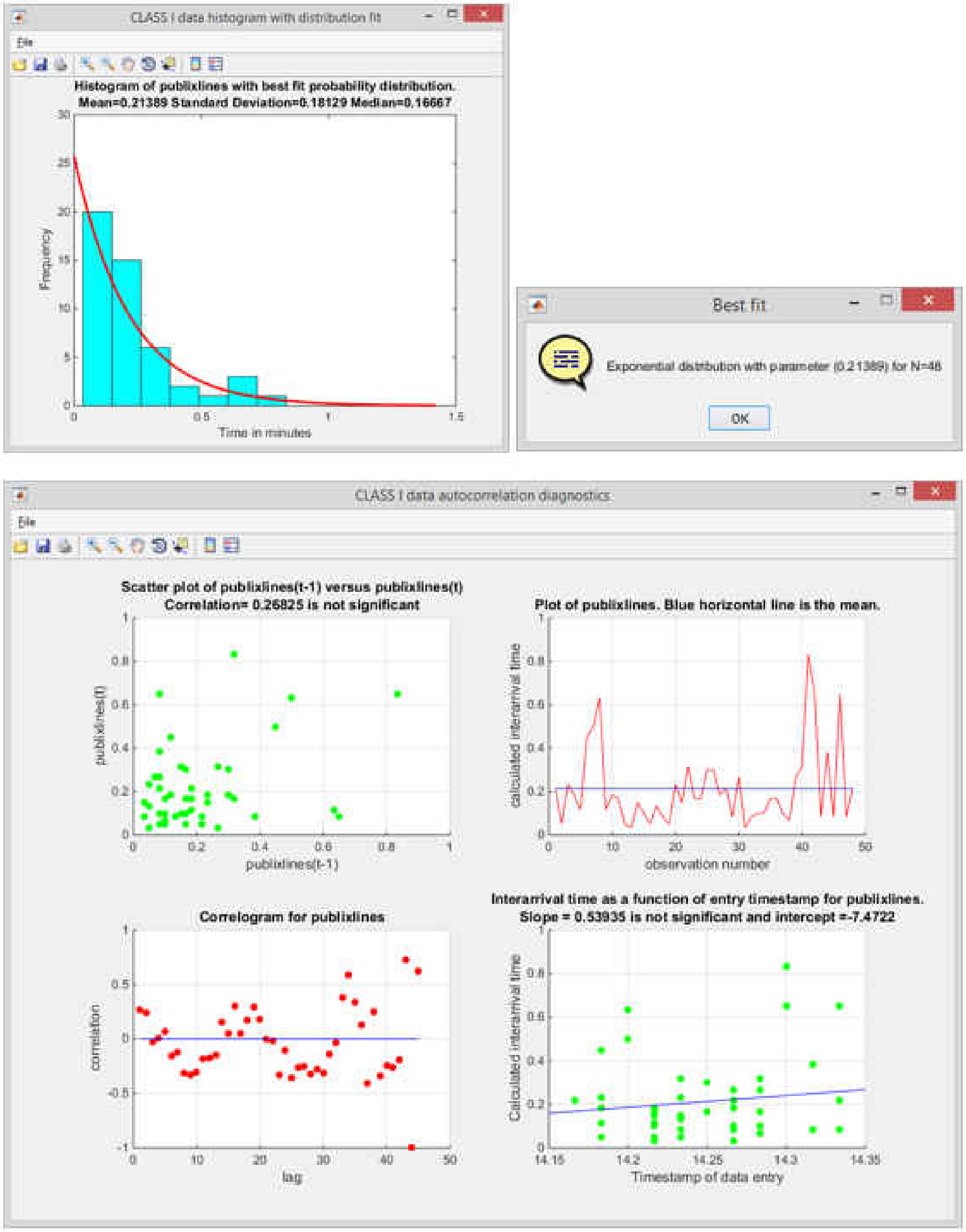
The Arena model for the Sanford International airport is presented in **Figure 50**. This model presents the utilization of the TSA officer checking travel documents and also the usage of the X-ray machine to check passengers. Utilization is represented by the green and white squares of **Figure 50**.The model logic includes the revision of traveling documents and the wait times before entering the X-ray machine. Model logic also includes the fact that passengers are retained downstairs until there is enough room in the upper level to accommodate queued passengers. 17% of this project's time was spent in input data management out of a total of 3 hours. More details about this project can be found in **Table 2**.

Lake Mary Publix Cashiers

This project was carried out at the Publix grocery store of Lake Mary, Florida. The goal of this project is to analyze the demand for cashier lanes on a Sunday, when there is high demand for the cashiers. This project was created with scheduled periods of full capacity, that is, when all cashiers are open to customers and scheduled periods when one lane is closed. The scenario modeled in this project is based on a Sunday afternoon's activity, with periods when one cashier was not active. The intention is analyzing the impact of opening an inactive cashier when there is high demand. In order to complete this project, the necessary data were:

- Data on IID arrivals into the cashiers collected using the blue section of DESI, as shown in **Figure 51**.
- Data on percentage clients going into fast cashier (10 items or less) collected using the green section of DESI as shown in **Figure 52**.
- IID intervals of service on a regular cashier collected using the orange section of DESI as shown in **Figure 53**.

- IID intervals of service on the fast lane cashier collected using the orange section of DESI as shown in **Figure 54**.


CLASS I data histogram with distribution fit
File
Histogram of publixlines with best fit probability distribution.
Mean=0.21389 Standard Deviation=0.18129 Median=0.16667
Frequency
Time in minutes
0
5
10
15
20
25
30
0.5
1
1.5
Best fit
OK
CLASS I data autocorrelation diagnostics
File
Scatter plot of publixlines(t-1) versus publixlines(t)
Correlation= 0.26825 is not significant
publixlines(t)
publixlines(t-1)
Plot of publixlines. Blue horizontal line is the mean.
calculated interarrival time
observation number
Correlogram for publixlines
correlation
lag
Interarrival time as a function of entry timestamp for publixlines.
Slope = 0.53935 is not significant and intercept =-7.4722
Calculated interarrival time
Timestamp of data entry
14.15
14.2
14.25
14.3
14.35

Exponential distribution with parameter (0.21389) for N=48

Figure 51 IID arrivals into Publix cashiers and autocorrelation diagnostics.

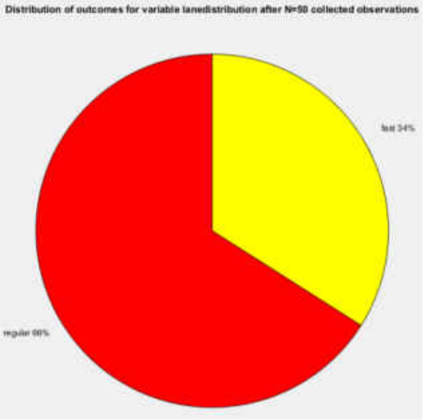
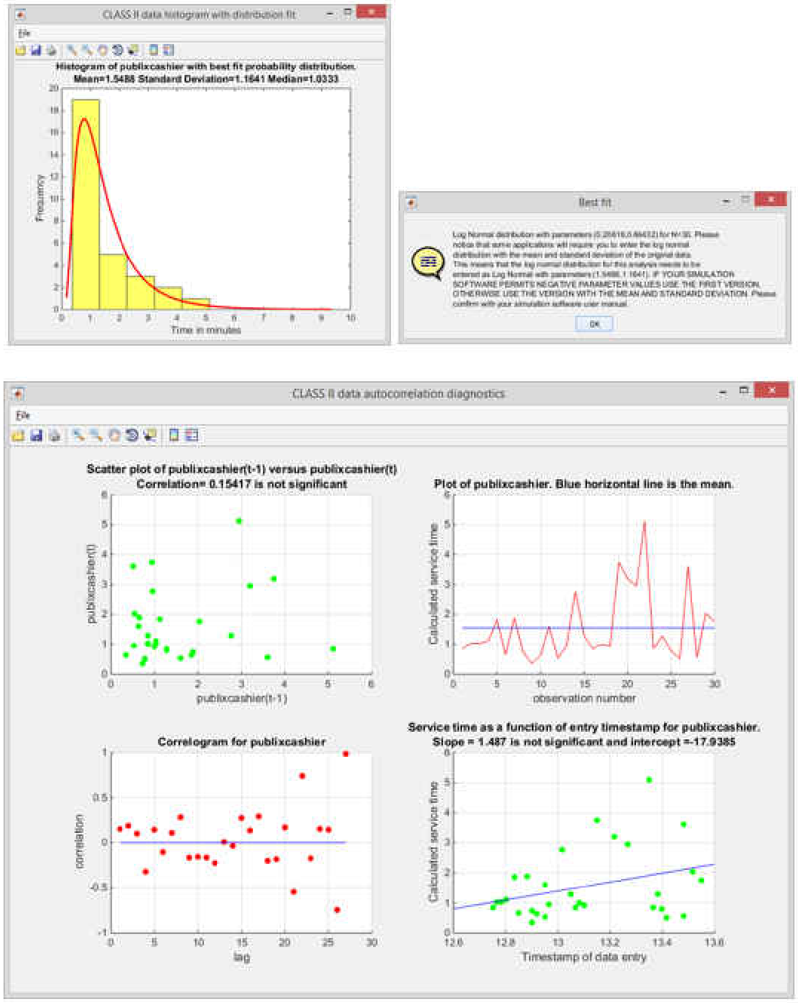
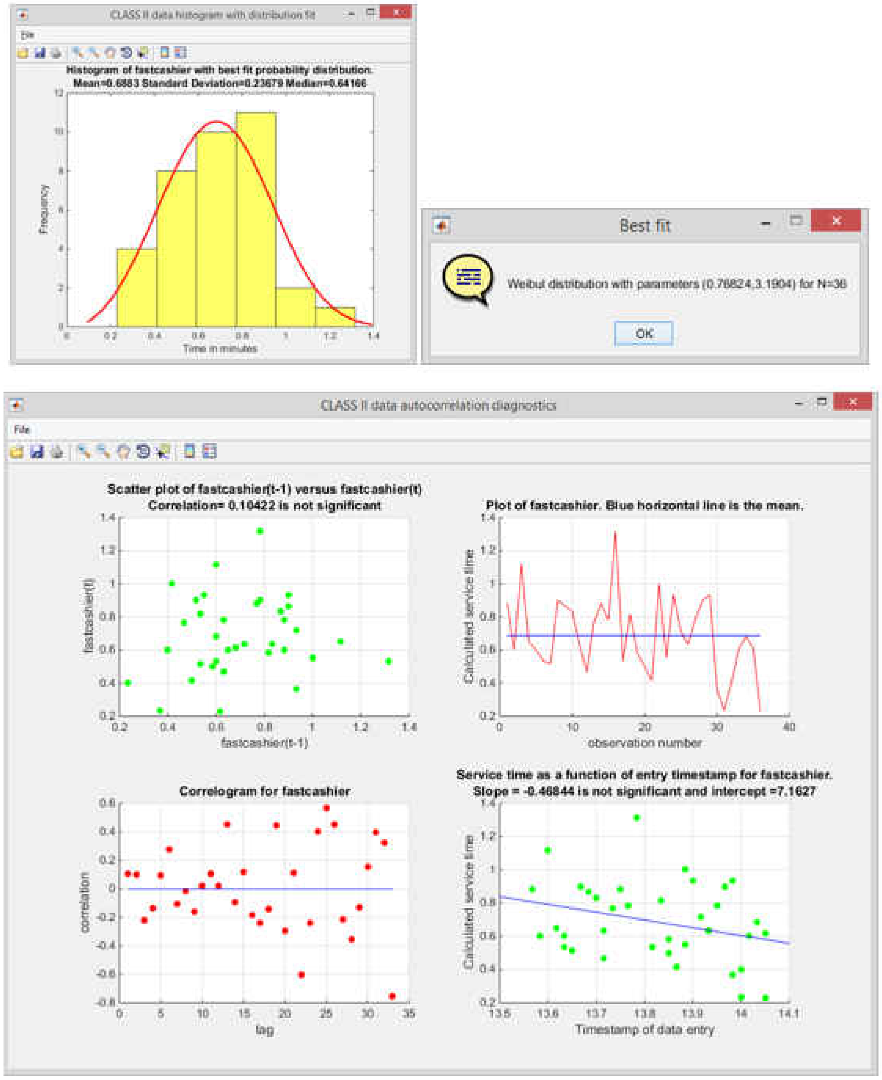


Figure 52 Percent of customers using the fast cashier


CLASS II data histogram with distribution fit
File
Histogram of publixcashier with best fit probability distribution.
Mean=1.5488 Standard Deviation=1.1641 Median=1.0333
Frequency
Time in minutes
Best fit
OK
CLASS II data autocorrelation diagnostics
File
Scatter plot of publixcashier(t-1) versus publixcashier(t)
Correlation= 0.15417 is not significant
publixcashier(t)
publixcashier(t-1)
Plot of publixcashier. Blue horizontal line is the mean.
Calculated service time
observation number
Correlogram for publixcashier
correlation
lag
Service time as a function of entry timestamp for publixcashier.
Slope = 1.487 is not significant and intercept =-17.9385
Calculated service time
Timestamp of data entry

Log Normal distribution with parameters (0.20616,0.66432) for N=30. Please notice that some applications will require you to enter the log normal distribution with the mean and standard deviation of the original data. This means that the log normal distribution for this analysis needs to be entered as Log Normal with parameters (1.5488,1.1641). IF YOUR SIMULATION SOFTWARE PERMITS NEGATIVE PARAMETER VALUES USE THE FIRST VERSION, OTHERWISE USE THE VERSION WITH THE MEAN AND STANDARD DEVIATION. Please confirm with your simulation software user manual.

Figure 53 IID Publix regular cashiers' service times and autocorrelation diagnostics.


CLASS II data histogram with distribution fit
File
Histogram of fastcashier with best fit probability distribution.
Mean=0.6883 Standard Deviation=0.23679 Median=0.64166
Frequency
0
2
4
6
8
10
12
0
0.2
0.4
0.6
0.8
1
1.2
1.4
Time in minutes
Best fit
OK
CLASS II data autocorrelation diagnostics
File
Scatter plot of fastcashier(t-1) versus fastcashier(t)
Correlation= 0.10422 is not significant
fastcashier(t)
fastcashier(t-1)
0.2
0.4
0.6
0.8
1
1.2
1.4
Plot of fastcashier. Blue horizontal line is the mean.
Calculated service time
observation number
0
10
20
30
40
Correlogram for fastcashier
correlation
lag
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0
5
10
15
20
25
30
35
Service time as a function of entry timestamp for fastcashier.
Slope = -0.46844 is not significant and intercept =7.1627
Calculated service time
Timestamp of data entry
13.5
13.6
13.7
13.8
13.9
14
14.1

Weibul distribution with parameters (0.76824,3.1904) for N=36

Figure 54 IID service times of Publix's fast cashier and autocorrelation diagnostics.

The Publix cashiers Arena Model is presented in **Figure 55**. The logic sends about 34% of customers to the fast cashier’s lane, based on collected data. It was also observed that if the fast lane had a long queue, customers leave that lane for a smaller lane if there is one. Entities that use the regular cashiers select the one with the smallest queue. **Figure 55** shows the utilization of the cashiers with green squares if the cashier is busy. The inactive cashier is represented by the white square with tilted lines. 30% of this project's time was spent in input data management out of a total of 5 hours. More details about this project can be found in **Table 2**.

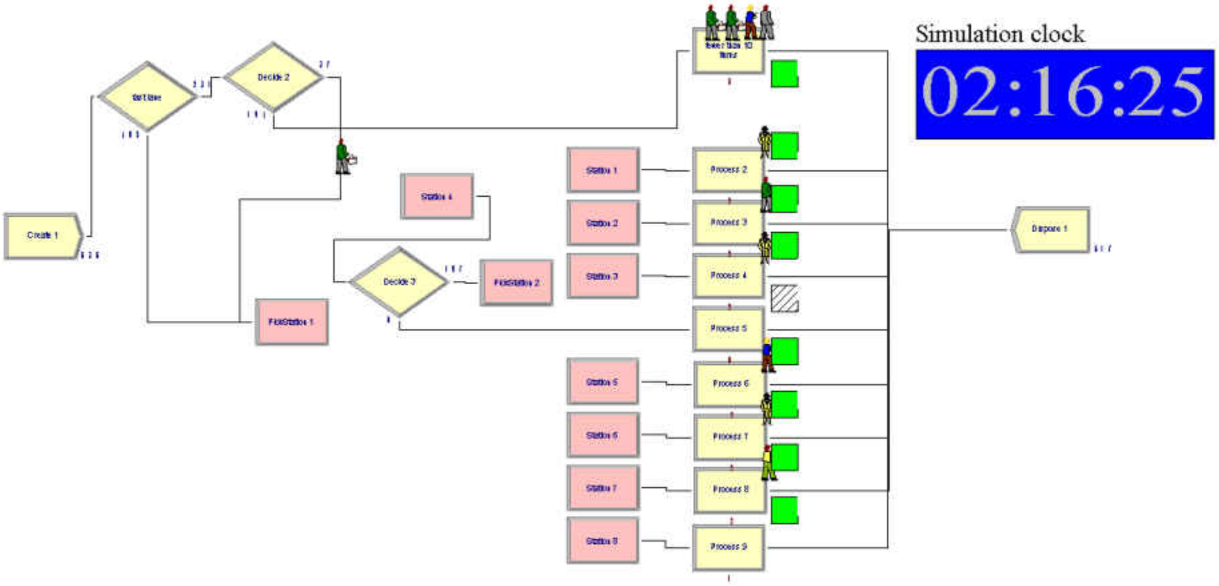


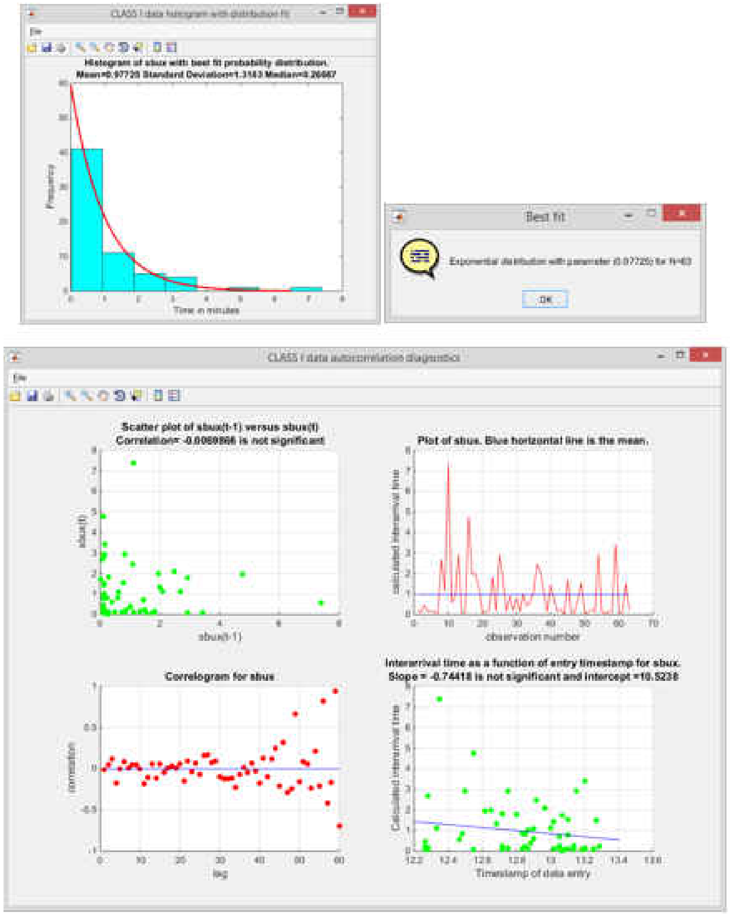
Figure 55 Publix cashiers Arena model.

Starbucks Winter Park Village

This project was carried out at the Starbucks coffee shop located at 600 N Orlando Ave, Winter Park, FL. This project focuses on Starbucks' personnel utilization during periods of high demand. Long waiting times during periods of high demand can deter the experience of customers. In order to complete this project, the necessary data were:

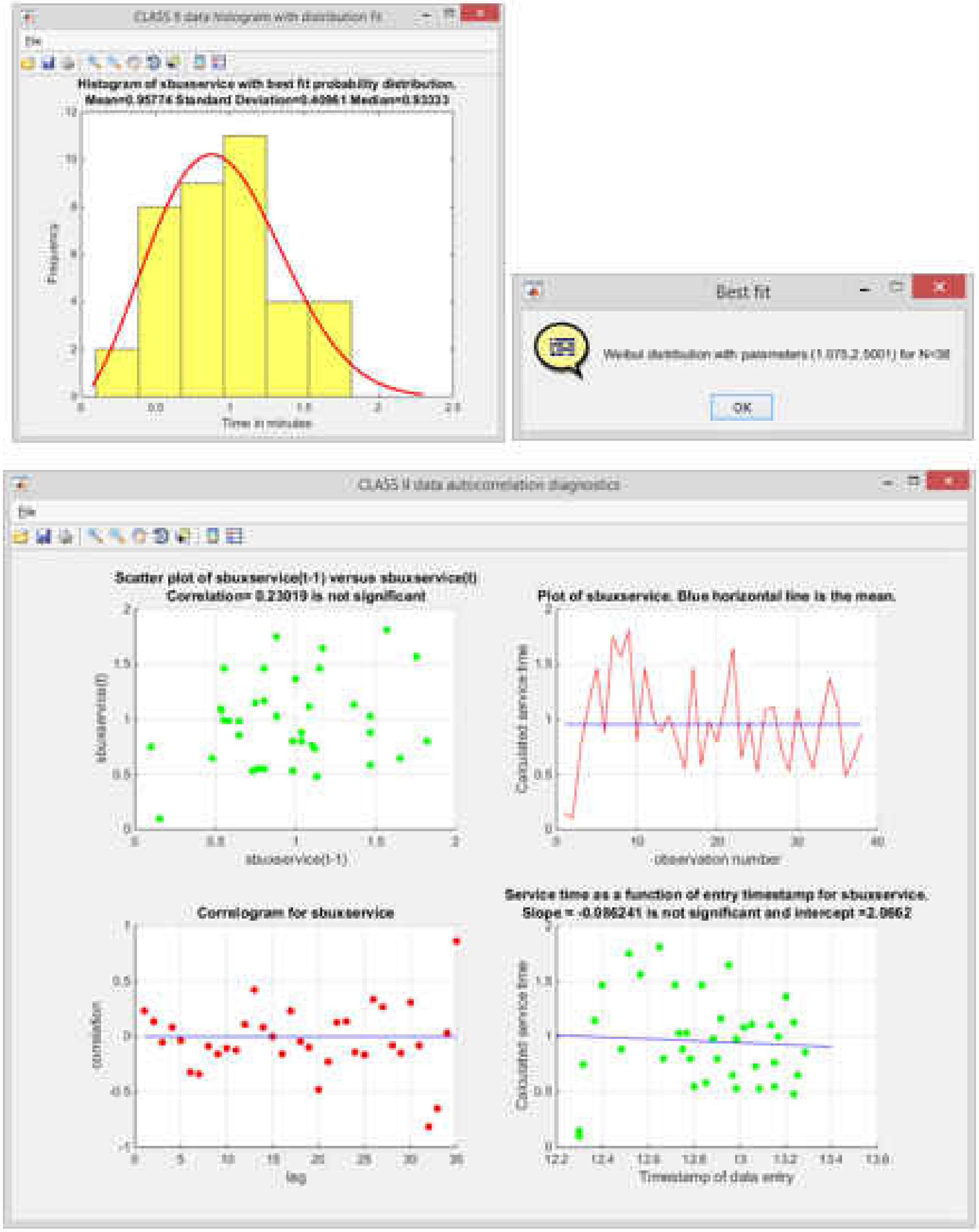
- IID arrivals into Starbucks gathered using the blue section of DESI and presented in **Figure 56**.
- IID payment times gathered the orange section of DESI and presented in **Figure 57**.

- IID order waiting times gathered using the orange section of DESI and presented in **Figure 58**.
- Percentage of times a customer is asked to wait for his or her order, gathered using the green section of DESI, as presented in **Figure 59**.
- Percentage of consumers who stay in the coffee shop gathered using the green section of DESI, as presented in **Figure 60**.
- Number of available tables and seats; once a party is in a table, the table is considered taken and other customers do not share the seized table. This data enters the model as part of the model logic.


CLASS I data histogram with distribution fit
File
Histogram of sbux with best fit probability distribution.
Mean=0.97725 Standard Deviation=1.3163 Median=0.26667
Frequency
Time in minutes
0
10
20
30
40
50
60
0
1
2
3
4
5
6
7
8
Best fit
OK
CLASS I data autocorrelation diagnostics
File
Scatter plot of sbux(t-1) versus sbux(t)
Correlation= -0.0069866 is not significant
sbux(t)
sbux(t-1)
Plot of sbux. Blue horizontal line is the mean.
Calculated interarrival time
observation number
Correlogram for sbux
correlation
lag
Interarrival time as a function of entry timestamp for sbux.
Slope = -0.74418 is not significant and intercept =10.5238
Calculated interarrival time
Timestamp of data entry
12.2
12.4
12.6
12.8
13
13.2
13.4
13.6

Exponential distribution with parameter (0.97725) for N=63

Figure 56 IID arrivals into Starbucks and autocorrelation diagnostics.



CLASS II data histogram with distribution fit

Histogram of sbuxservice with best fit probability distribution.
Mean=0.95774 Standard Deviation=0.40961 Median=0.93333

Best fit

Weibull distribution with parameters (1.075,2.5001) for N=38

OK

CLASS II data autocorrelation diagnostics

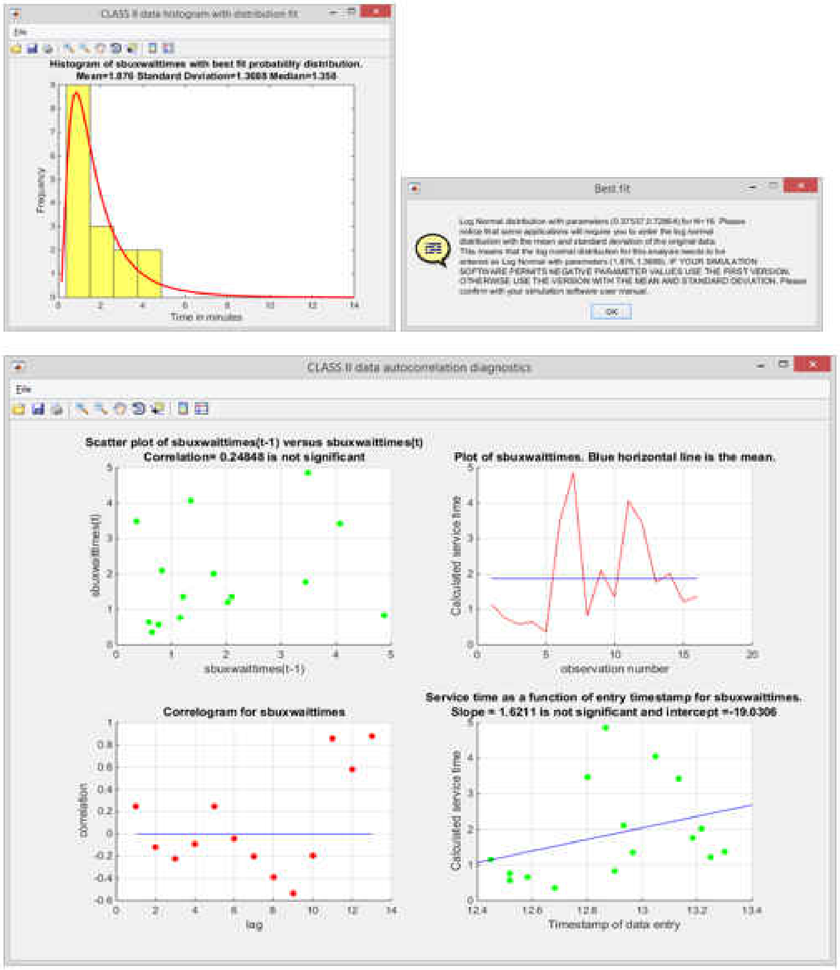
Scatter plot of sbuxservice(t-1) versus sbuxservice(t)
Correlation= 0.23019 is not significant

Plot of sbuxservice. Blue horizontal line is the mean.

Correlogram for sbuxservice

Service time as a function of entry timestamp for sbuxservice.
Slope = -0.086241 is not significant and intercept =2.0662

Figure 57 IID payment times in Starbucks cashiers and autocorrelation diagnostics.


CLASS II data histogram with distribution fit
File
Histogram of sbuxwaittimes with best fit probability distribution.
Mean=1.876 Standard Deviation=1.3688 Median=1.358
Frequency
Time in minutes
0
2
4
6
8
10
12
14
Best fit

Log Normal distribution with parameters (0.37537,0.72864) for N=16. Please notice that some applications will require you to enter the log normal distribution with the mean and standard deviation of the original data. This means that the log normal distribution for this analysis needs to be entered as Log Normal with parameters (1.876,1.3688). IF YOUR SIMULATION SOFTWARE PERMITS NEGATIVE PARAMETER VALUES USE THE FIRST VERSION, OTHERWISE USE THE VERSION WITH THE MEAN AND STANDARD DEVIATION. Please confirm with your simulation software user manual.

OK
CLASS II data autocorrelation diagnostics
File
Scatter plot of sbuxwaittimes(t-1) versus sbuxwaittimes(t)
Correlation= 0.24848 is not significant
sbuxwaittimes(t)
sbuxwaittimes(t-1)
Plot of sbuxwaittimes. Blue horizontal line is the mean.
Calculated service time
observation number
Correlogram for sbuxwaittimes
correlation
lag
Service time as a function of entry timestamp for sbuxwaittimes.
Slope = 1.6211 is not significant and intercept =-19.0306
Calculated service time
Timestamp of data entry
12.4
12.6
12.8
13
13.2
13.4

Figure 58 IID data on Starbucks order completion and autocorrelation diagnostics.

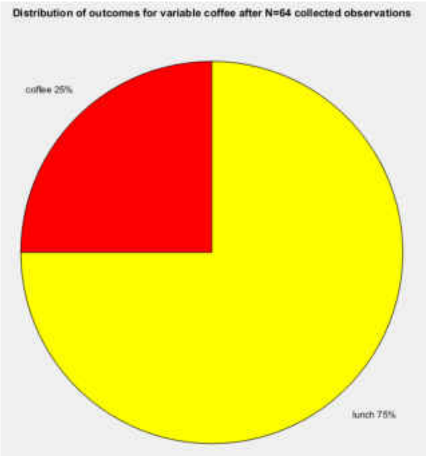


Figure 59 Percentage of customers who order black coffee.

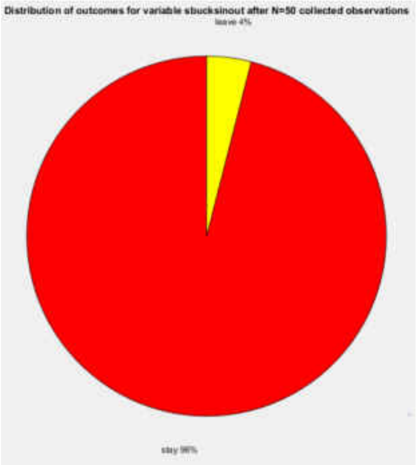


Figure 60 Percentage of customers who stay in coffee shop.

The Starbucks coffee shop Arena model is displayed in **Figure 61**. This model focuses on table utilization and staff utilization during lunch hours, when employees of businesses around the coffee shop coincide in the coffee shop in addition to regular Starbucks customers. The model presents busy resources with green squares. Resources include staff and tables available for seating. Even though some tables can seat more than one person, usually when a person or group of persons take a table, no one else sits on it. 30% of this project's time was spent in input data management out of a total of 5 hours. Details about this project can be found in **Table 2**.

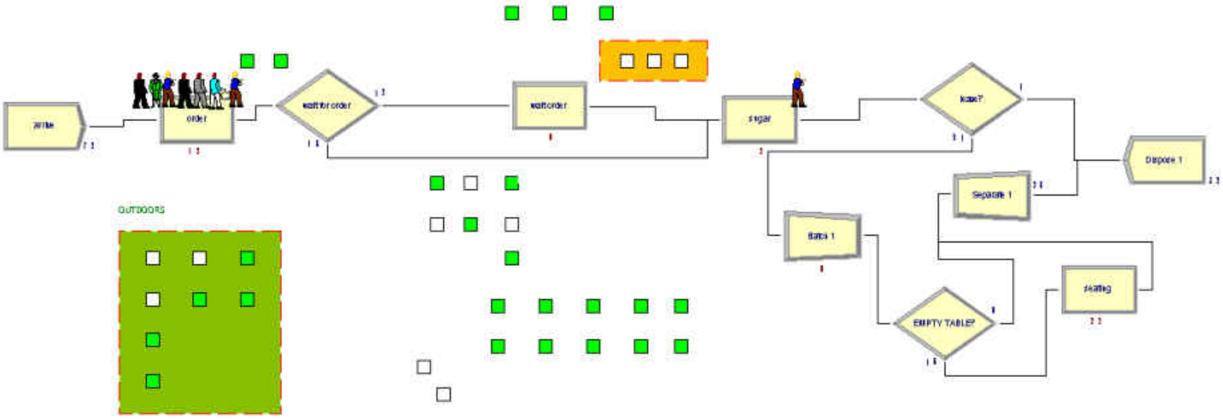
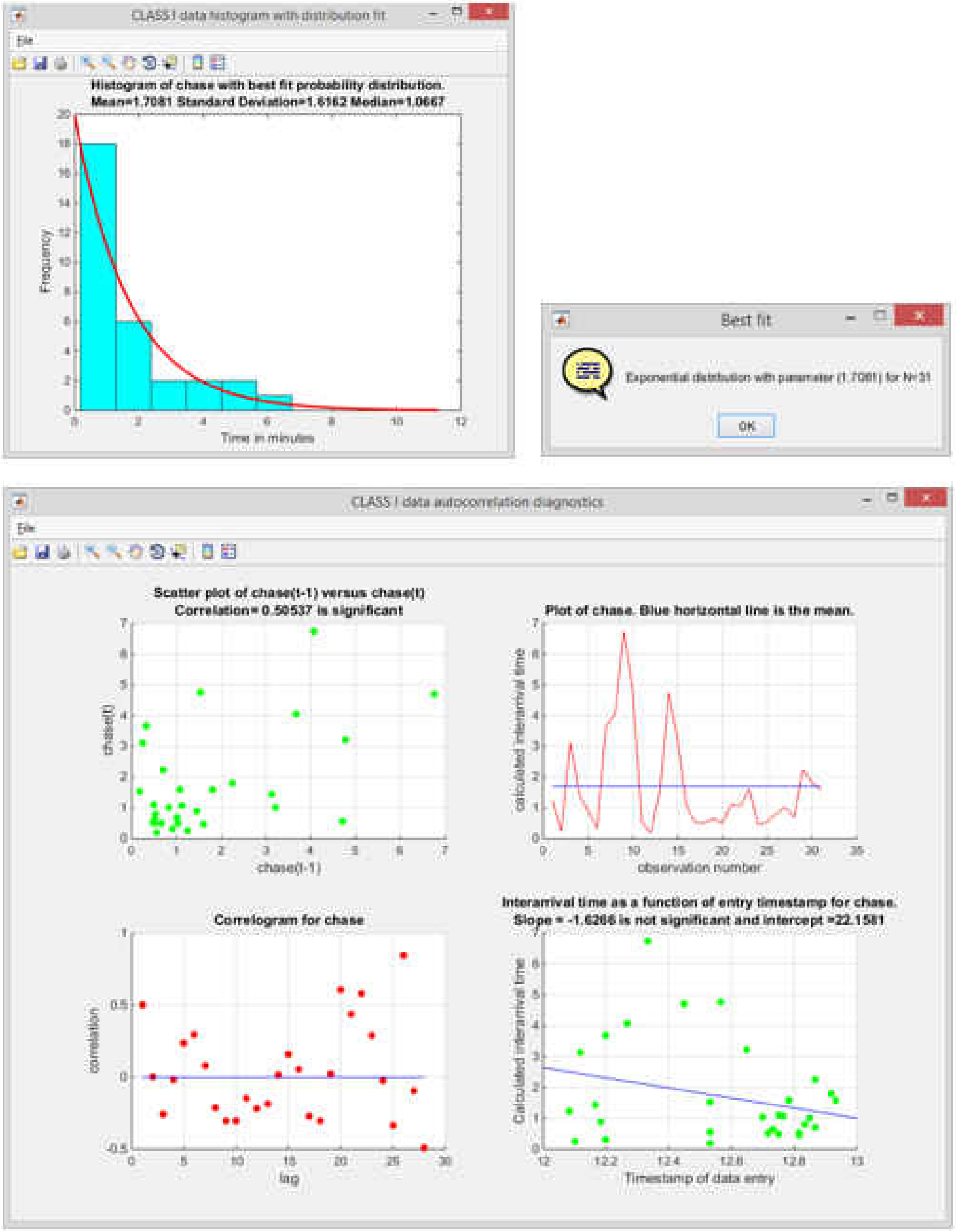


Figure 61 Starbucks coffee shop Arena model.

Winter Park Village Chase Bank Drive Thru

This project was carried out at the Chase Bank located at 600 N Orlando Ave, Winter Park, FL. This project analyzes the traffic in the bank's drive thru stations and the drive thru ATM. In order to complete this project, the necessary data were:

- Arrivals into the Chase Bank drive thru using the blue section of DESI and presented in **Figure 62**. This is an example on how DESI can help detect autocorrelation. Notice the significant serial correlation. The simulation project was still carried out, but this example shows the usefulness of DESI in validating DES theoretical assumptions.
- IID service times at the drive thru banking stations collected using the orange section of DESI and presented in **Figure 63**.
- Percentage of times a customer uses the ATM drive thru station versus other banking services, collected using the green section of DESI and presented in **Figure 64**.
- IID data on ATM drive thru station usage collected using the orange section of DESI and presented in **Figure 65**.



Histogram of chase with best fit probability distribution.
Mean=1.7081 Standard Deviation=1.6162 Median=1.0667

Exponential distribution with parameter (1.7081) for N=31

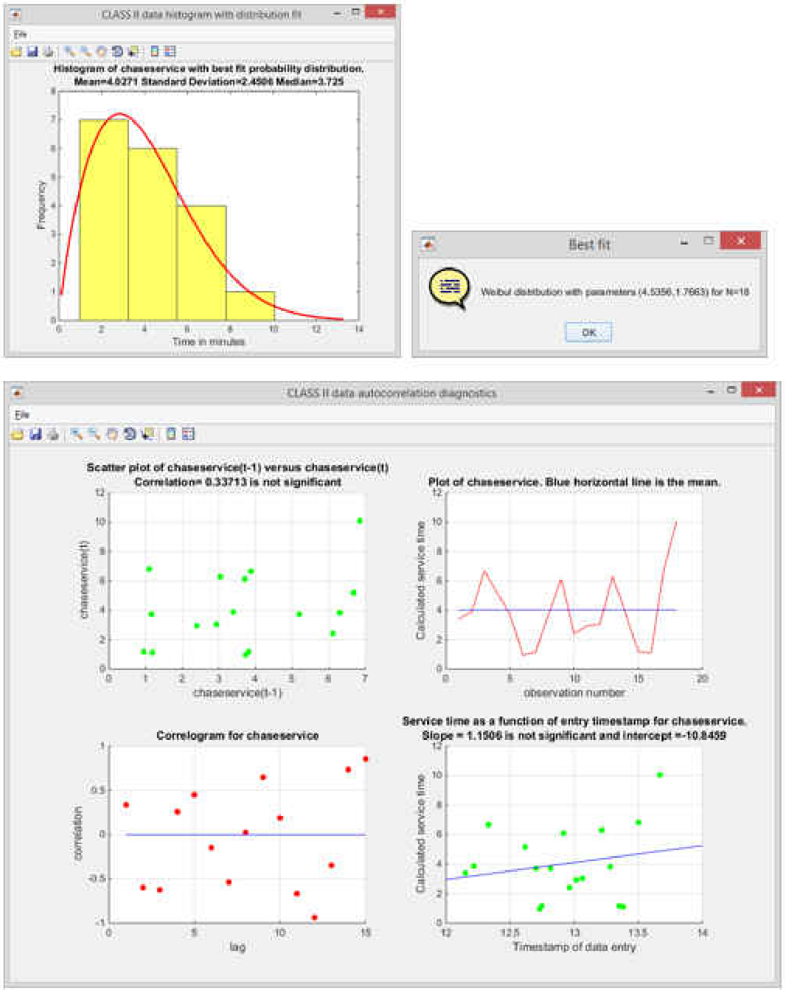
Scatter plot of chase(t-1) versus chase(t)
Correlation= 0.50537 is significant

Plot of chase. Blue horizontal line is the mean.

Correlogram for chase

Interarrival time as a function of entry timestamp for chase.
Slope = -1.6266 is not significant and intercept =22.1581

Figure 62 Automated detection of serial correlation using DESI.


CLASS II data histogram with distribution fit
File
Histogram of chaseservice with best fit probability distribution.
Mean=4.0271 Standard Deviation=2.4506 Median=3.725
Frequency
0
1
2
3
4
5
6
7
8
0
2
4
6
8
10
12
14
Time in minutes
Best fit
OK
CLASS II data autocorrelation diagnostics
File
Scatter plot of chaseservice(t-1) versus chaseservice(t)
Correlation= 0.33713 is not significant
chaseservice(t)
chaseservice(t-1)
0
1
2
3
4
5
6
7
0
2
4
6
8
10
12
Plot of chaseservice. Blue horizontal line is the mean.
Calculated service time
observation number
0
5
10
15
20
Correlogram for chaseservice
correlation
lag
1
0.5
0
-0.5
-1
0
5
10
15
Service time as a function of entry timestamp for chaseservice.
Slope = 1.1506 is not significant and intercept =-10.8459
Calculated service time
Timestamp of data entry
12
12.5
13
13.5
14

Weibul distribution with parameters (4.5356,1.7663) for N=18

Figure 63 IID Chase drive thru banking times.

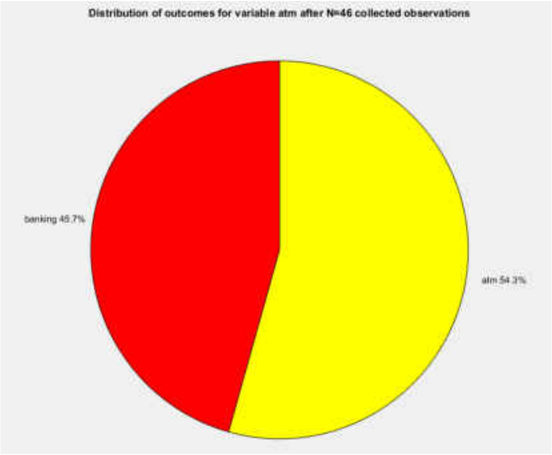
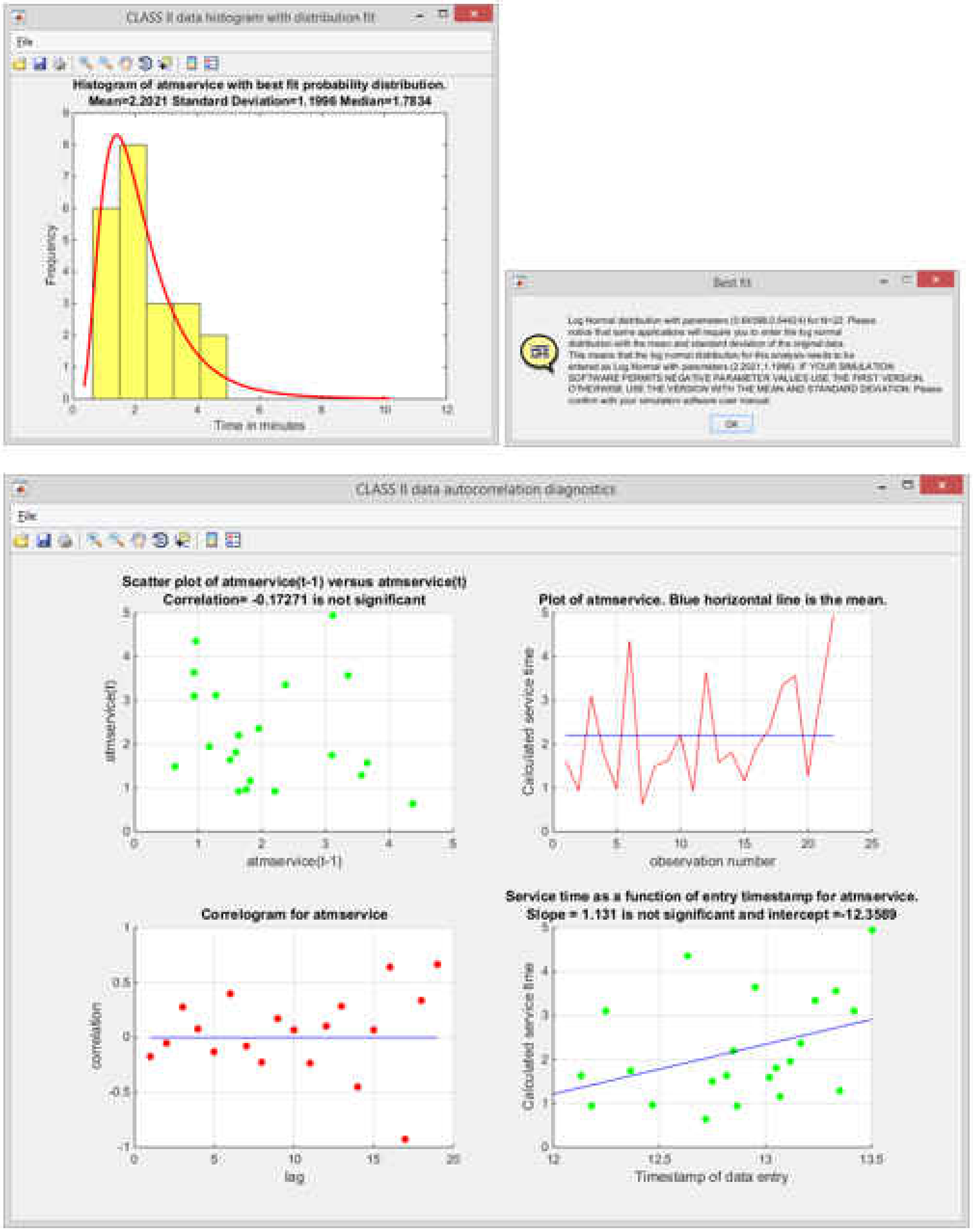


Figure 64 Percentage of cars going into Chase bank drive thru ATM.


CLASS II data histogram with distribution fit
File
Histogram of atmservice with best fit probability distribution.
Mean=2.2021 Standard Deviation=1.1996 Median=1.7834
Frequency
Time in minutes
0
1
2
3
4
5
6
7
8
9
10
12
Best fit
OK
CLASS II data autocorrelation diagnostics
File
Scatter plot of atmservice(t-1) versus atmservice(t)
Correlation= -0.17271 is not significant
atmservice(t)
atmservice(t-1)
Plot of atmservice. Blue horizontal line is the mean.
Calculated service time
observation number
Correlogram for atmservice
correlation
lag
Service time as a function of entry timestamp for atmservice.
Slope = 1.131 is not significant and intercept =-12.3589
Calculated service time
Timestamp of data entry
0
5
10
15
20
25
-1
-0.5
0.5
12
12.5
13
13.5

Log Normal distribution with parameters (0.60598,0.54424) for N=22. Please notice that some applications will require you to enter the log normal distribution with the mean and standard deviation of the original data. This means that the log normal distribution for this analysis needs to be entered as Log Normal with parameters (2.2021,1.1996). IF YOUR SIMULATION SOFTWARE PERMITS NEGATIVE PARAMETER VALUES USE THE FIRST VERSION, OTHERWISE USE THE VERSION WITH THE MEAN AND STANDARD DEVIATION. Please confirm with your simulation software user manual.

Figure 65 IID Chase drive thru ATM usage times.

The Chase Bank drive thru Arena model is presented in **Figure 66**. The logic of this model allows incoming cars to determine which station they want to go into, depending on the queue size, unless customers only need the ATM. Resources are represented by bank staff and the ATM machine. Resource utilization is represented by green and white squares, depending on whether the resources are busy or idle. 43% of this project was devoted to input data management out of a total of 3.5 hours.

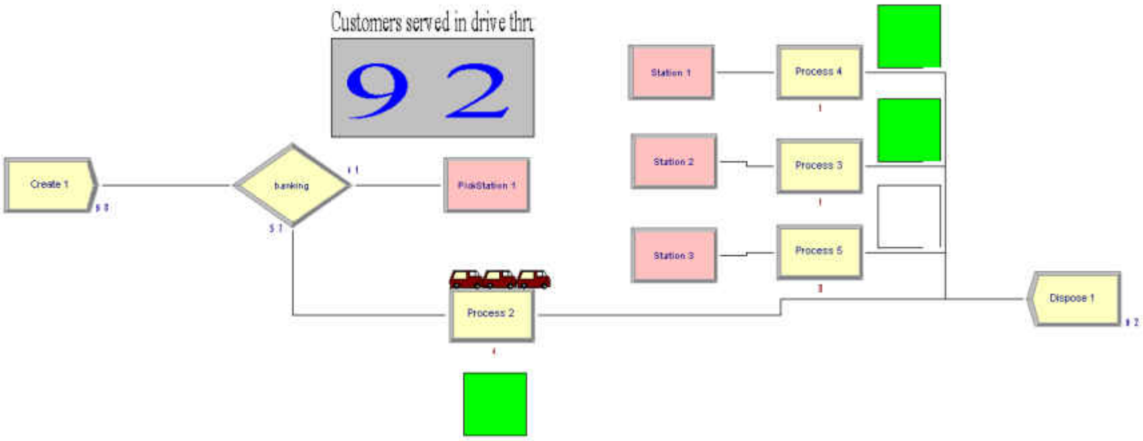


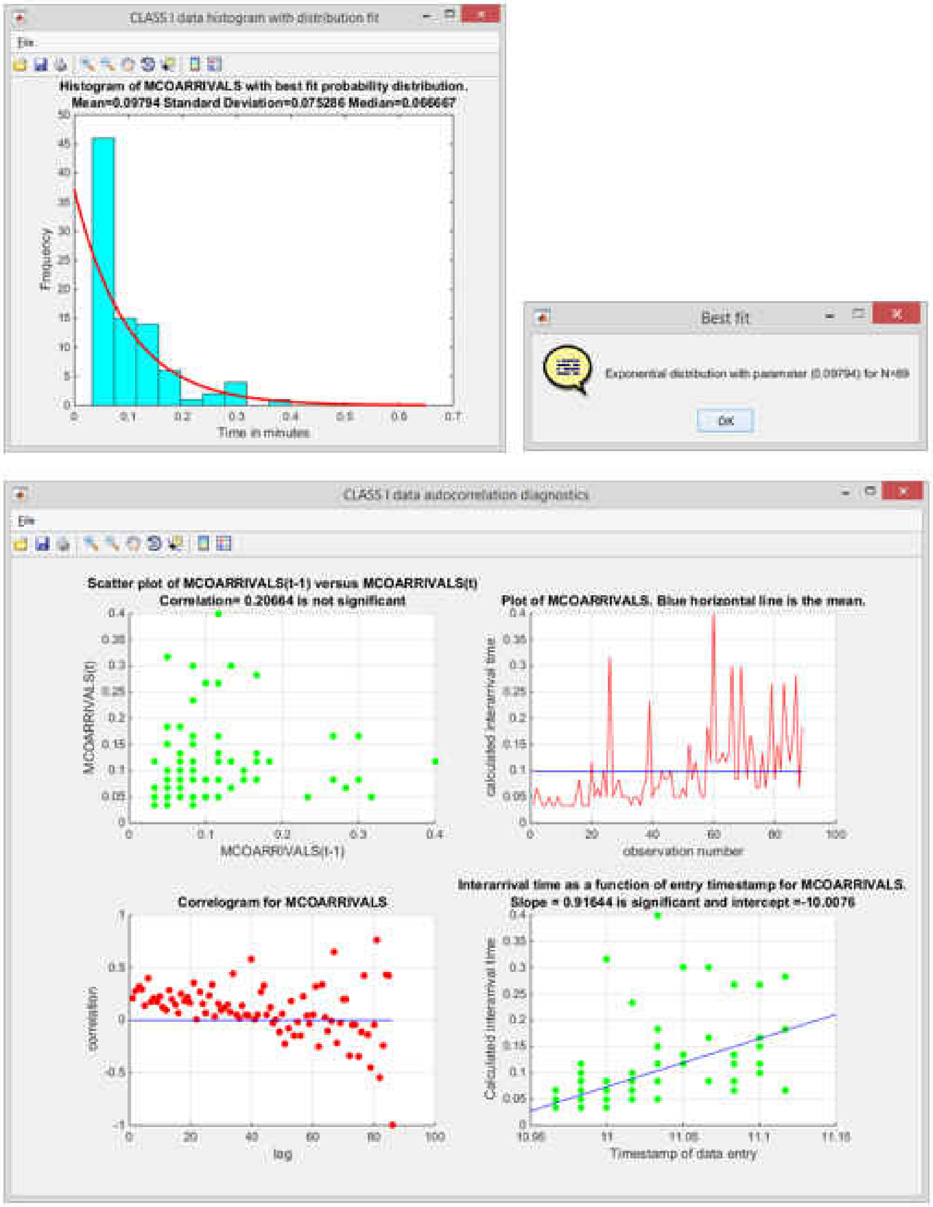
Figure 66 Chase bank drive thru Arena model.

Orlando International Airport (MCO) Security Checkpoint

This project was carried out at the Orlando International Airport in Orlando, Florida and it analyzes the flow of passengers thru the 10 boarding pass check stations and thru the 8 X-ray machines. The focus is on the security checkpoint for gates 1 to 59. In order to complete this project, the necessary data were:

- IID arrivals of passengers to the security check point gathered using the blue section of DESI and presented in **Figure 67**.
- IID times for boarding pass revision gathered using the orange section of DESI and presented in **Figure 68**.

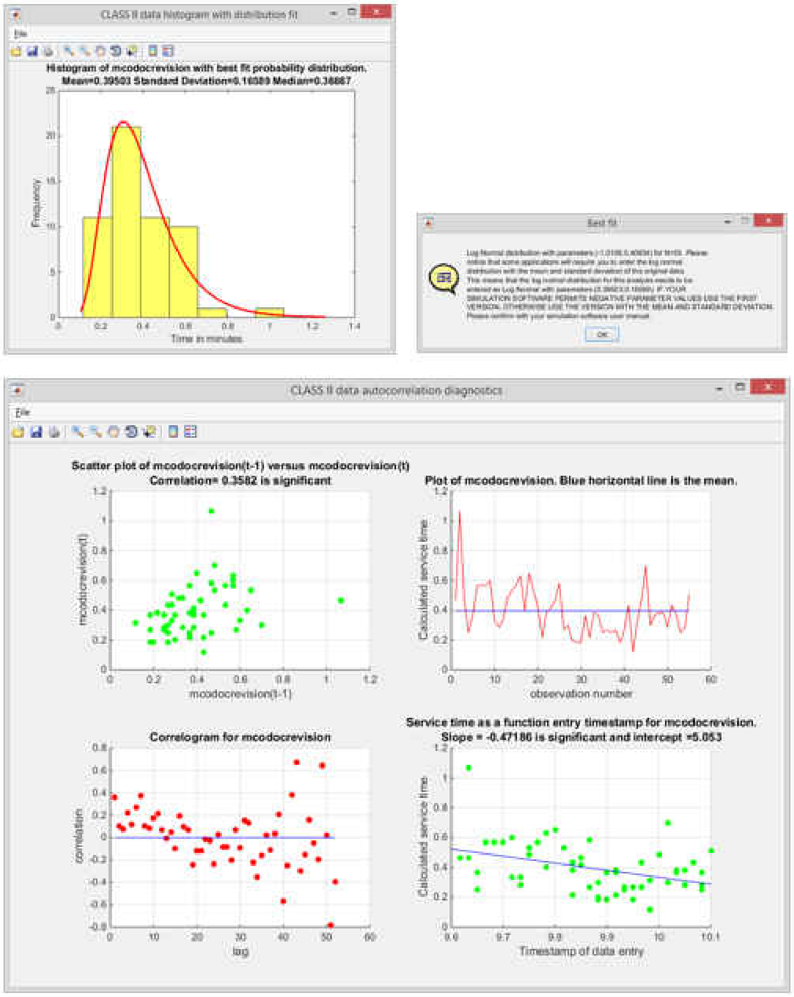
- Distribution of number of passengers arriving into the security checkpoint gathered using the green section of DESI and presented in **Figure 69**.
- IID wait times before using the X-ray machine gathered using the orange section of DESI and presented in **Figure 70**.



CLASS I data histogram with distribution fit
File
Histogram of MCOARRIVALS with best fit probability distribution.
Mean=0.09794 Standard Deviation=0.075286 Median=0.066667
Frequency
0
5
10
15
20
25
30
35
40
45
50
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Time in minutes
Best fit
OK
CLASS I data autocorrelation diagnostics
File
Scatter plot of MCOARRIVALS(t-1) versus MCOARRIVALS(t)
Correlation= 0.20664 is not significant
MCOARRIVALS(t)
MCOARRIVALS(t-1)
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0
0.1
0.2
0.3
0.4
Plot of MCOARRIVALS. Blue horizontal line is the mean.
Calculated interarrival time
observation number
0
20
40
60
80
100
Correlogram for MCOARRIVALS
correlation
lag
-1
-0.5
0
0.5
1
0
20
40
60
80
100
Interarrival time as a function of entry timestamp for MCOARRIVALS.
Slope = 0.91644 is significant and intercept =-10.0076
Calculated interarrival time
Timestamp of data entry
10.95
11
11.05
11.1
11.15

Exponential distribution with parameter (0.09794) for N=89

Figure 67 IID arrivals into MCO security check point and diagnostics.



Histogram of mcodocrevision with best fit probability distribution.
Mean=0.39503 Standard Deviation=0.16089 Median=0.38887

Log Normal distribution with parameters (-1.0108,0.40634) for N=55. Please notice that some applications will require you to enter the log normal distribution with the mean and standard deviation of the original data. This means that the log normal distribution for this analysis needs to be entered as Log Normal with parameters (0.39503,0.16089). IF YOUR SIMULATION SOFTWARE PERMITS NEGATIVE PARAMETER VALUES USE THE FIRST VERSION, OTHERWISE USE THE VERSION WITH THE MEAN AND STANDARD DEVIATION. Please confirm with your simulation software user manual.

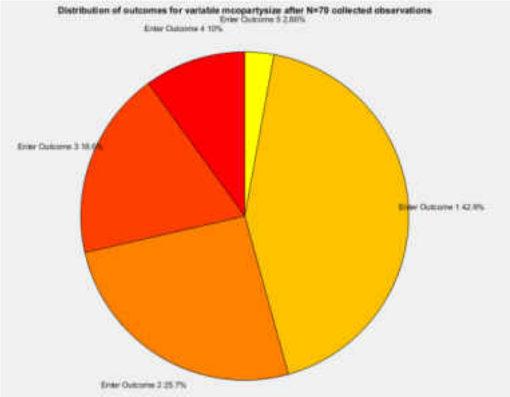
Scatter plot of mcodocrevision(t-1) versus mcodocrevision(t)
Correlation= 0.3582 is significant

Plot of mcodocrevision. Blue horizontal line is the mean.

Correlogram for mcodocrevision

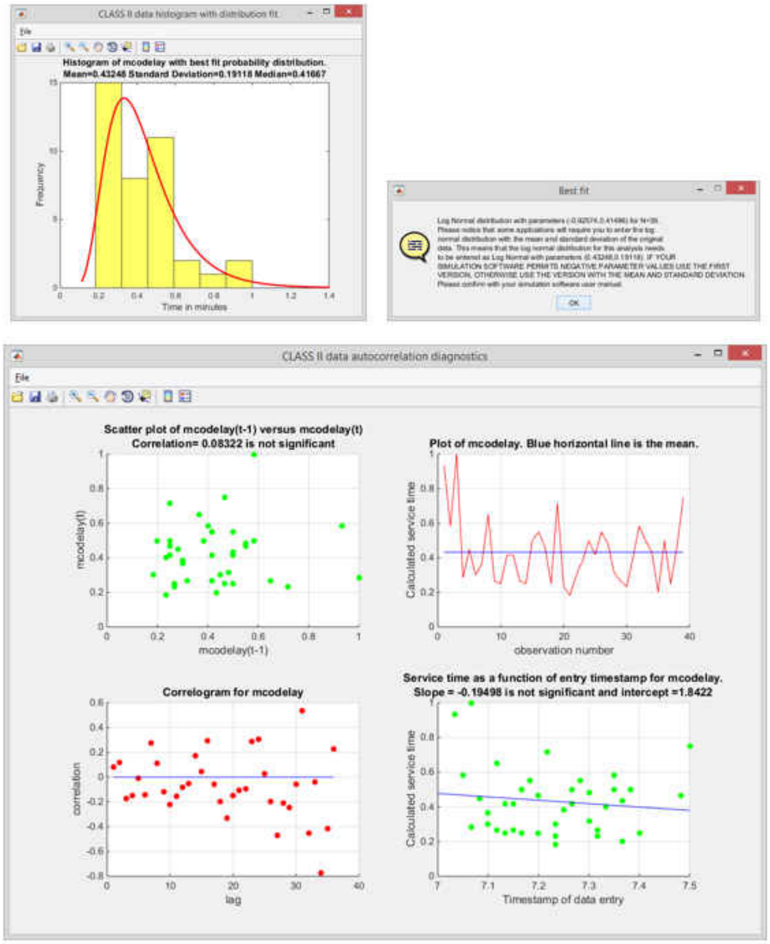
Service time as a function entry timestamp for mcodocrevision.
Slope = -0.47186 is significant and intercept =5.053

Figure 68 IID MCO revision of travel documents and autocorrelation diagnostics.



Distribution of outcomes for variable mcopartysize after N=70 collected observations

Figure 69 Distribution of group sizes entering security at Orlando Intl. Airport.


CLASS II data histogram with distribution fit
File
Histogram of mcodelay with best fit probability distribution.
Mean=0.43248 Standard Deviation=0.19118 Median=0.41667
Frequency
Time in minutes
Best fit
OK
CLASS II data autocorrelation diagnostics
File
Scatter plot of mcodelay(t-1) versus mcodelay(t)
Correlation= 0.08322 is not significant
mcodelay(t)
mcodelay(t-1)
Plot of mcodelay. Blue horizontal line is the mean.
Calculated service time
observation number
Correlogram for mcodelay
correlation
lag
Service time as a function of entry timestamp for mcodelay.
Slope = -0.19498 is not significant and intercept =1.8422
Calculated service time
Timestamp of data entry

Log Normal distribution with parameters (-0.92574,0.41496) for N=39.
Please notice that some applications will require you to enter the log-
normal distribution with the mean and standard deviation of the original
data. This means that the log normal distribution for this analysis needs
to be entered as Log Normal with parameters (0.43248,0.19118). IF YOUR
SIMULATION SOFTWARE PERMITS NEGATIVE PARAMETER VALUES USE THE FIRST
VERSION, OTHERWISE USE THE VERSION WITH THE MEAN AND STANDARD DEVIATION.
Please confirm with your simulation software user manual.

Figure 70 IID wait times before X-ray machines in MCO and diagnostics.

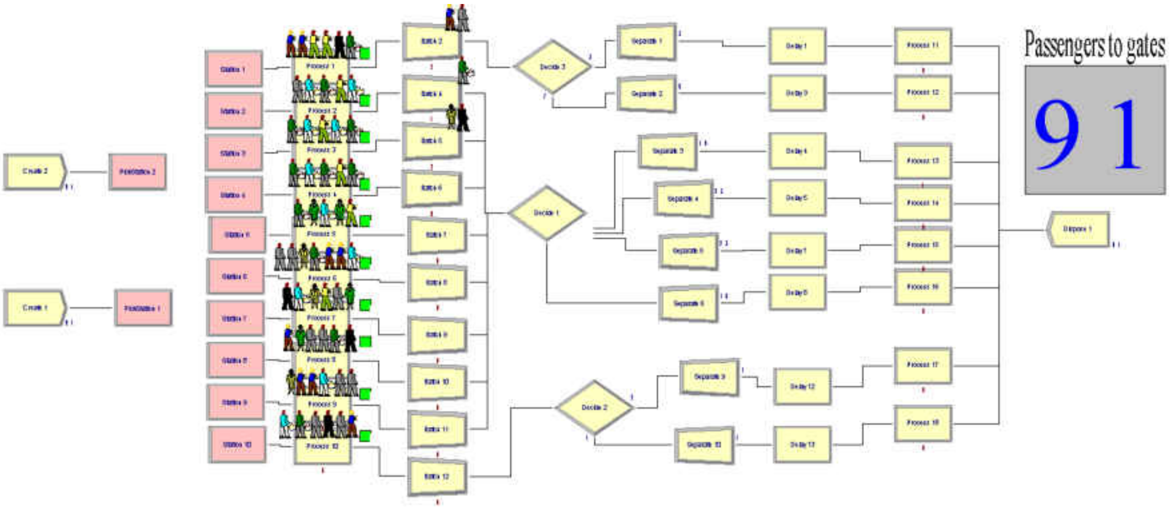


Figure 71 Orlando International Airport security Arena model.

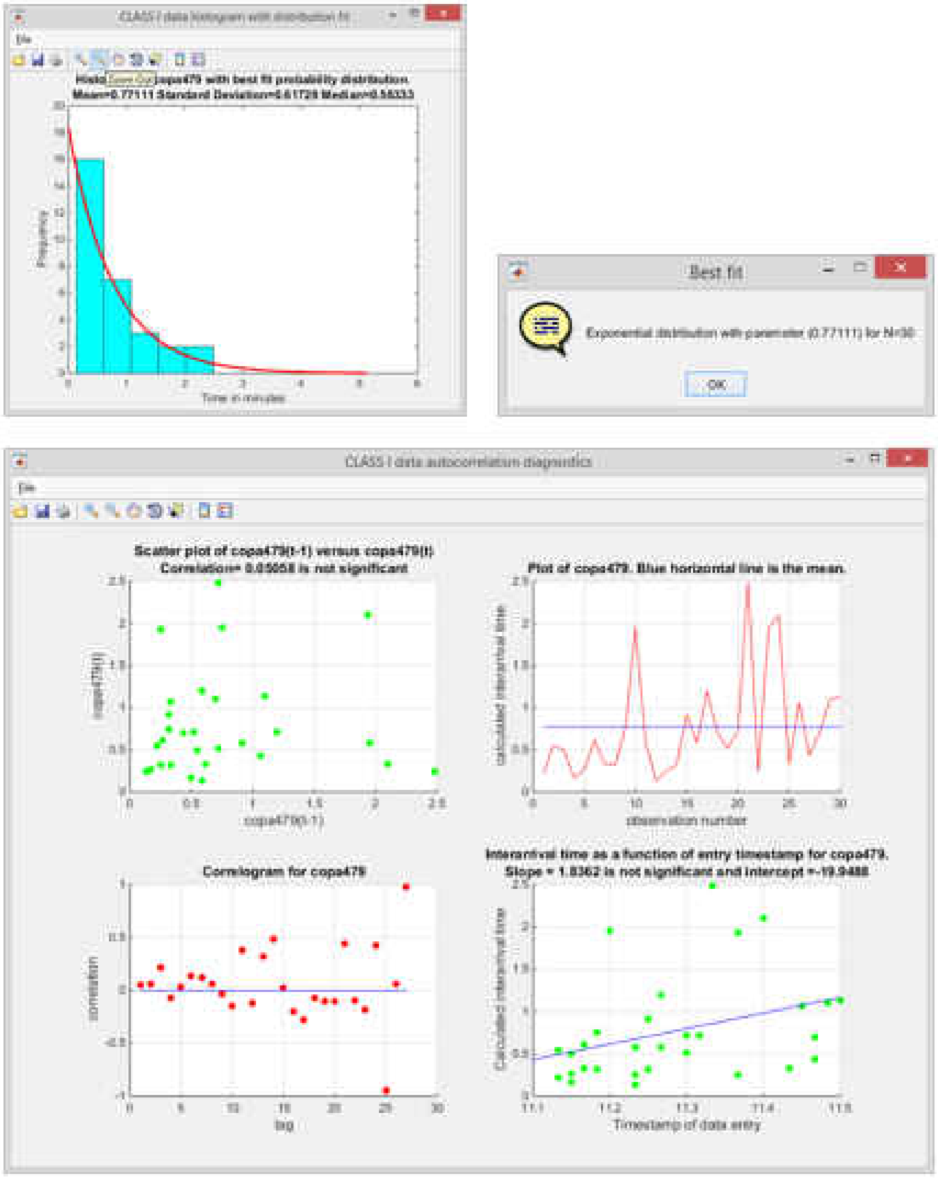
The Orlando International Airport Arena model is presented in **Figure 71**. The model logic features the passenger arrivals, who decide on which security lane to enter based on queue size. There are two entry points into security, and each point has 5 TSA agents checking travel documents. This project takes into account traveling group sizes, which are batched based on the probabilities obtained in the data collection process. The batching happens in order to direct families thru the same X-ray machine. Once a machine has been assigned, the batches are separated. High priority travelers have not been defined in this model, since the number of these travelers is negligible compared to the overall number of passengers, making almost no difference in the simulation results. 8% of this project's time was spent in input data management out of a total of 8 hours. More details about this project can be found in **Table 2**.

COPA Airlines Passenger Check-In Process

This project was carried out at the Orlando International Airport in Orlando, Florida and it analyzes the flow of passengers flying in COPA Airlines to Panama City, Panama. In order to complete this project, the necessary data were:

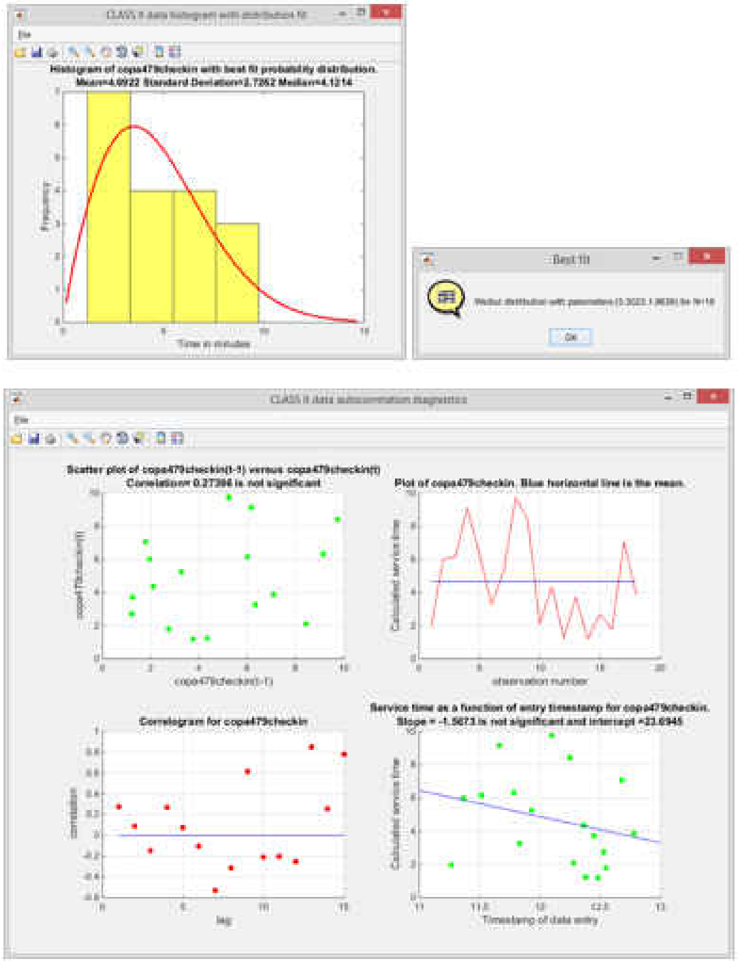
- IID arrivals of passengers to the check-in stations using the blue section of DESI and presented in **Figure 72**.
- IID times for the check-in process gathered using the orange section of DESI and presented in **Figure 73**.
- Distribution of number of passengers arriving into COPA Airlines check-in stations using the green section of DESI and presented in **Figure 74**.

- Distribution of the type of passengers (regular, premier, web check-in) arriving into COPA Airlines check-in stations using the green section of DESI and presented in **Figure 75**.


CLASS I data histogram with distribution fit
File
Histogram of copa479 with best fit probability distribution.
Mean=0.77111 Standard Deviation=0.61729 Median=0.58333
Frequency
Time in minutes
0
2
4
6
8
10
12
14
16
18
20
0
1
2
3
4
5
6
Best fit
OK
CLASS I data autocorrelation diagnostics
File
Scatter plot of copa479(t-1) versus copa479(t)
Correlation= 0.05058 is not significant
copa479(t)
copa479(t-1)
Plot of copa479. Blue horizontal line is the mean.
Calculated interarrival time
observation number
Correlogram for copa479
correlation
lag
Interarrival time as a function of entry timestamp for copa479.
Slope = 1.8362 is not significant and intercept =-19.9488
Calculated interarrival time
Timestamp of data entry
11.1
11.2
11.3
11.4
11.5

Exponential distribution with parameter (0.77111) for N=30

Figure 72 IID passenger arrivals into COPA Airlines' check-in and diagnostics.


CLASS II data histogram with distribution fit
File
Histogram of copa479checkin with best fit probability distribution.
Mean=4.0922 Standard Deviation=2.7262 Median=4.1214
Frequency
Time in minutes
0
5
10
15
Best fit
OK
CLASS II data autocorrelation diagnostics
File
Scatter plot of copa479checkin(t-1) versus copa479checkin(t)
Correlation= 0.27386 is not significant
copa479checkin(t)
copa479checkin(t-1)
Plot of copa479checkin. Blue horizontal line is the mean.
Calculated service time
observation number
Correlogram for copa479checkin
correlation
lag
Service time as a function of entry timestamp for copa479checkin.
Slope = -1.5673 is not significant and intercept =23.6945
Calculated service time
Timestamp of data entry

Weibull distribution with parameters [5.3023 1.8639] for N=18

Figure 73 IID COPA Airlines check-in times and autocorrelation diagnostics.

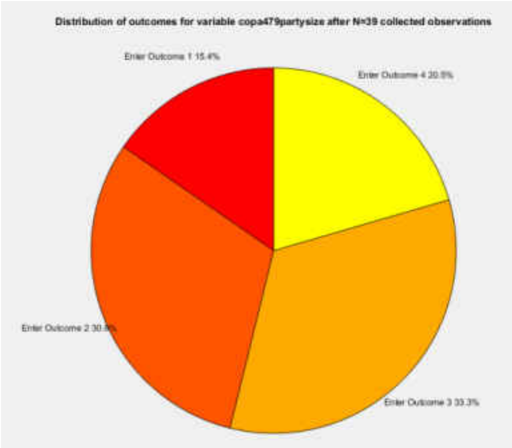


Figure 74 Group size distribution arriving for COPA check-in.

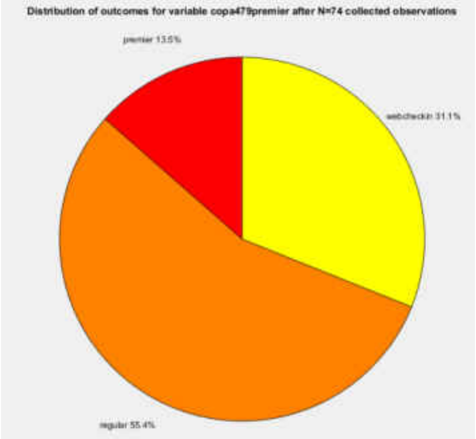


Figure 75 Passenger type distribution arriving for COPA check-in.

The COPA Airlines Arena model is presented in **Figure 76**. The model logic sends the entities to the open check-in stations where COPA staff provides boarding passes. Family sizes are represented by batches of entities, which need to be sent to the same station. Staff member utilization is presented by green squares in **Figure 76**. Passenger priority and group sizes are assigned based on probability. Then signals for open stations are sent in order to release entities so they can seize the open stations based on availability. 20% of this project's time was spent in input data management out of a total of 7.5 hours. More details about this project can be found in **Table 2**.

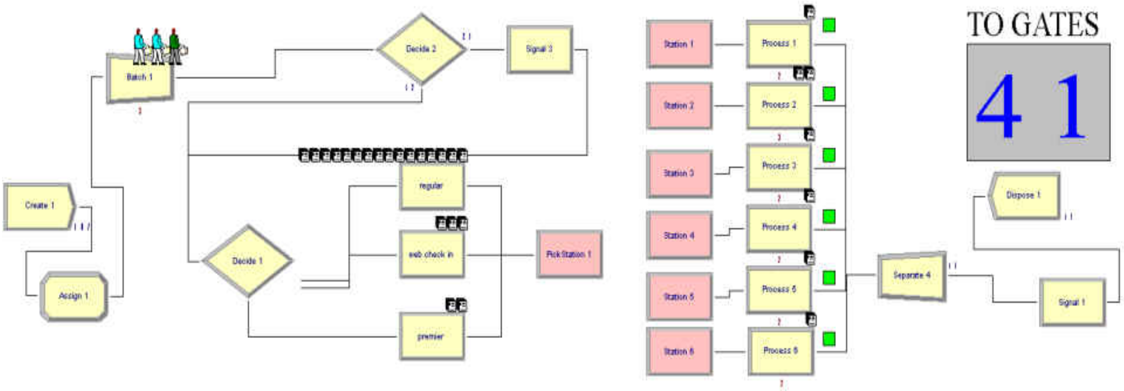


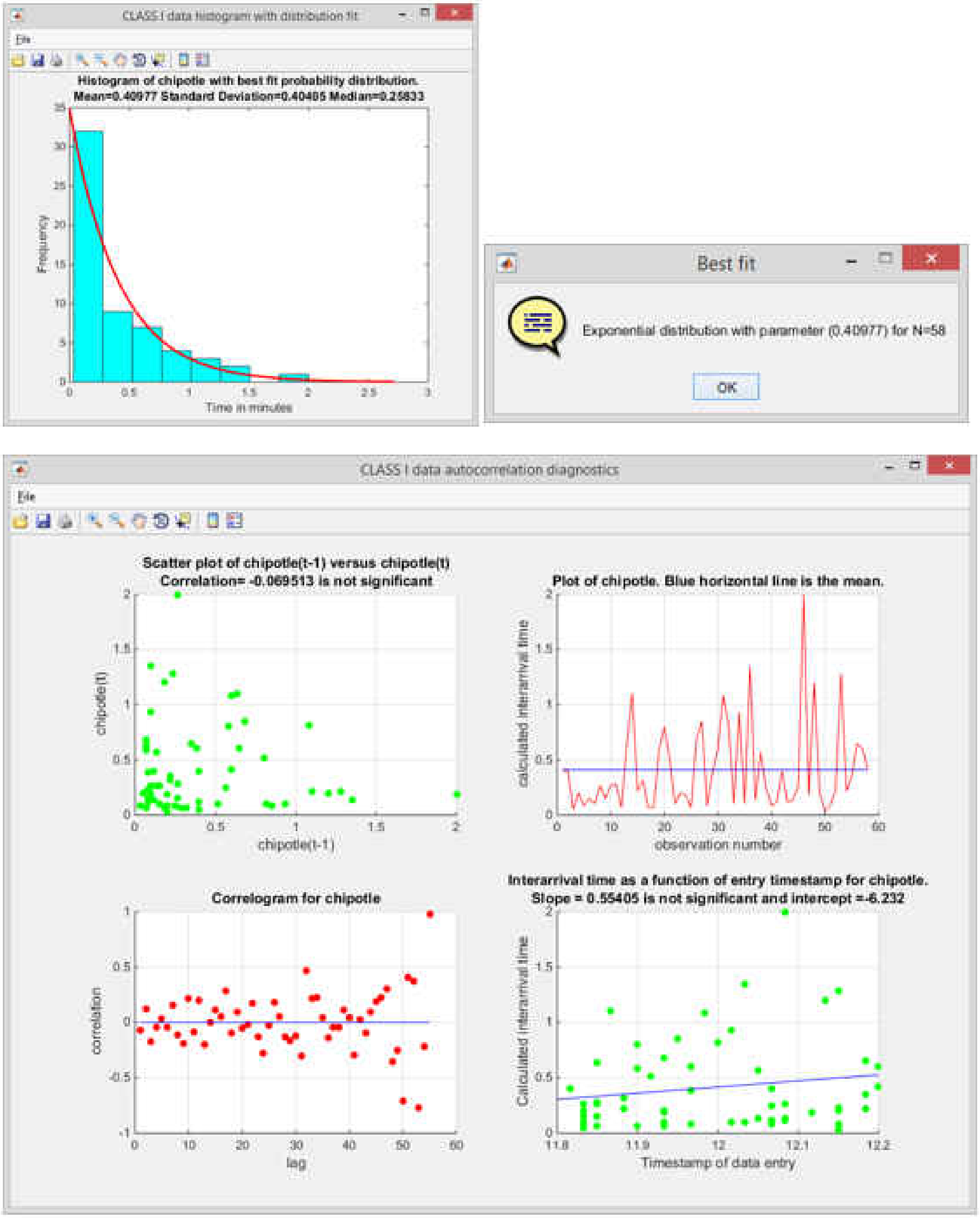
Figure 76 COPA Airlines check-in Arena model.

Chipotle Mexican Grill

This project was carried out at the Chipotle Mexican Grill in the intersection of Orlando Avenue and Fairbanks Avenue in Orlando, Florida and it analyzes the heavy demand during lunch hours. In order to complete this project, the necessary data were:

- IID arrivals of customers gathered using the blue section of DESI and presented in **Figure 77**.
- Ordering times gathered using the orange section of DESI and presented **Figure 78**. This is another instance of how DESI can help detect serial correlation.
- IID cashier serving times gathered using the orange section of DESI and presented in **Figure 79**.
- Distribution of number of customers arriving to Chipotle gathered using the green section of DESI and presented in **Figure 80**.

The Chipotle Arena model is displayed in **Figure81**. This model logic prevents entities from being served if there is no more room in the serving area. Idle employees, other than the ones taking orders and receiving payments, send signals to different hold modules in order to release one entity at a time so the one entity can be served. Busy Chipotle employees receiving the orders and getting the payments are represented by green squares in **Figure 81**. The flow of customers is determined by the paying times, which must be fast in order to prevent queues to end up forming outside the restaurant. 10% of this project's time was spent in input data management out of a total of 5 hours. More details about this project can be found in **Table 2**.



CLASS I data histogram with distribution fit

Histogram of chipotle with best fit probability distribution.
Mean=0.40977 Standard Deviation=0.40405 Median=0.25833

Best fit

Exponential distribution with parameter (0.40977) for N=58

CLASS I data autocorrelation diagnostics

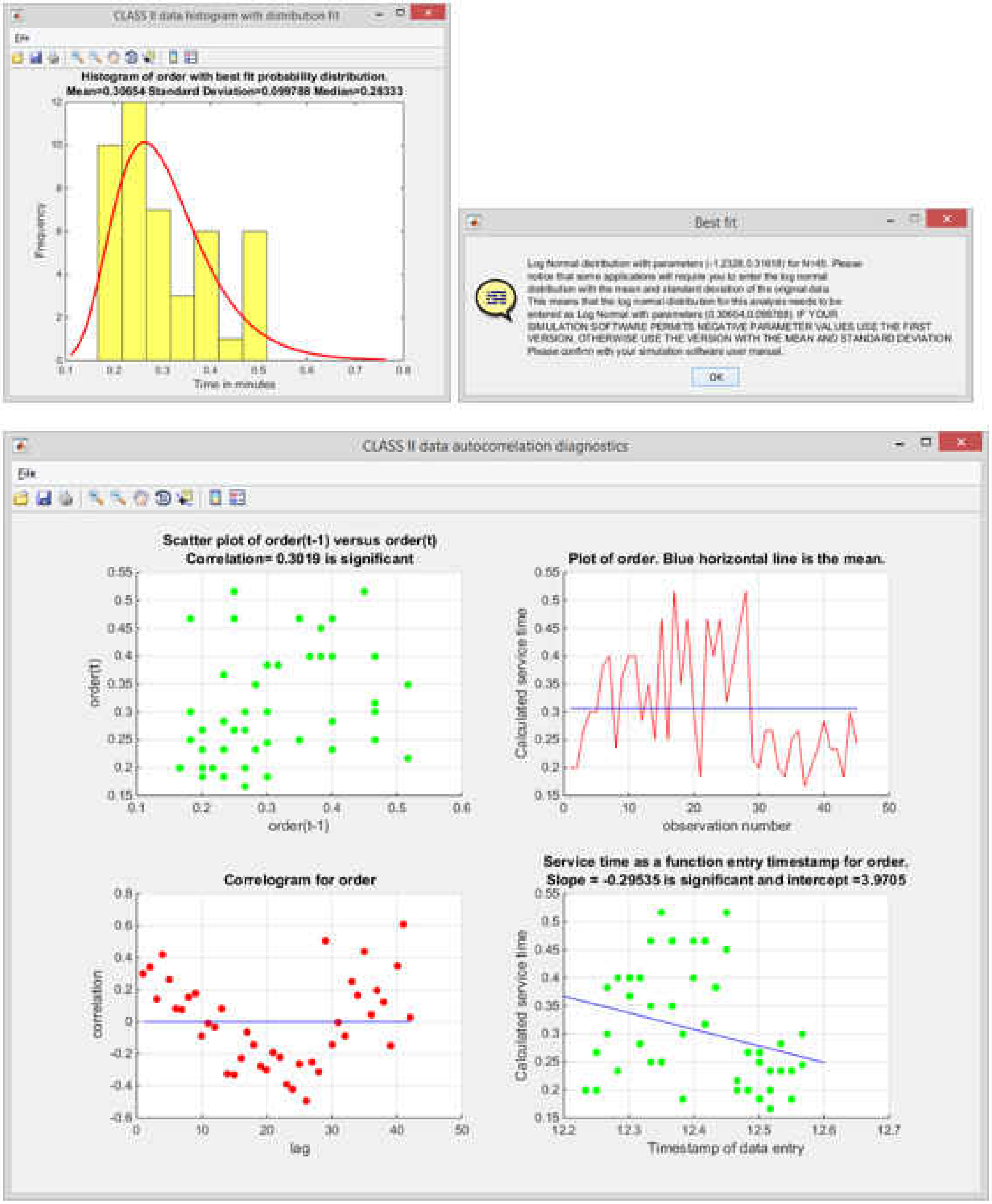
Scatter plot of chipotle(t-1) versus chipotle(t)
Correlation= -0.069513 is not significant

Plot of chipotle. Blue horizontal line is the mean.

Correlogram for chipotle

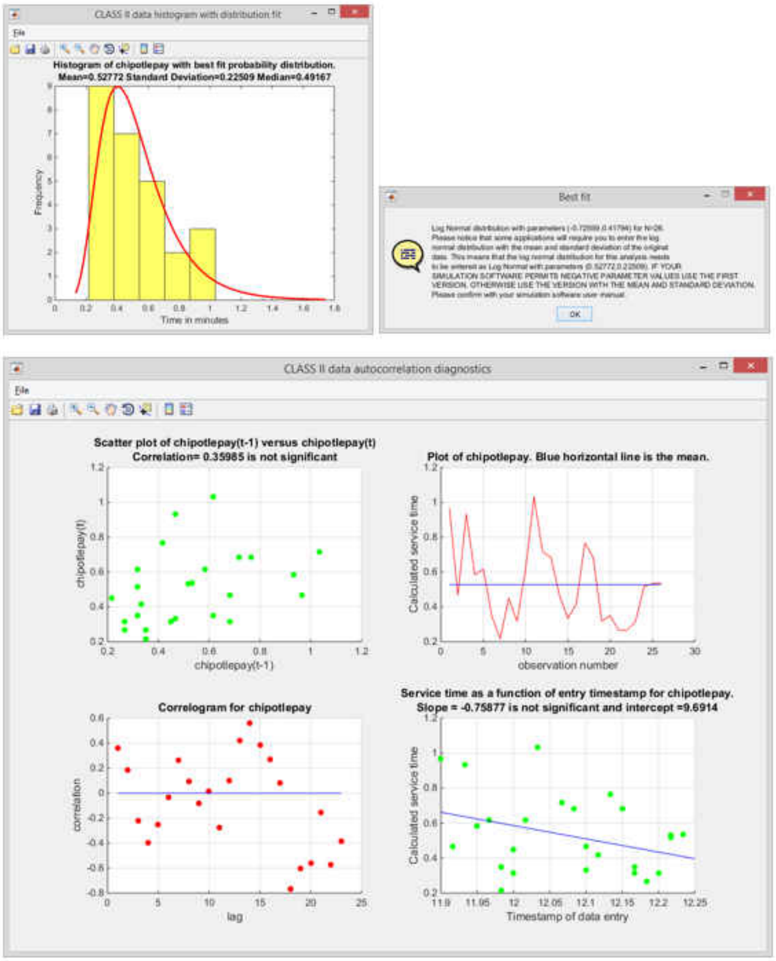
Interarrival time as a function of entry timestamp for chipotle.
Slope = 0.55405 is not significant and intercept =-6.232

Figure 77 IID arrivals into Chipotle and autocorrelation diagnostics.


CLASS II data histogram with distribution fit
File
Histogram of order with best fit probability distribution.
Mean=0.30654 Standard Deviation=0.099788 Median=0.28333
Frequency
Time in minutes
0
2
4
6
8
10
12
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Best fit
OK
CLASS II data autocorrelation diagnostics
File
Scatter plot of order(t-1) versus order(t)
Correlation= 0.3019 is significant
order(t)
order(t-1)
Plot of order. Blue horizontal line is the mean.
Calculated service time
observation number
Correlogram for order
correlation
lag
Service time as a function entry timestamp for order.
Slope = -0.29535 is significant and intercept =3.9705
Calculated service time
Timestamp of data entry
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55
0.1
0.2
0.3
0.4
0.5
0.6
0
10
20
30
40
50
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
12.2
12.3
12.4
12.5
12.6
12.7

Log Normal distribution with parameters (-1.2328,0.31918) for N=45. Please notice that some applications will require you to enter the log normal distribution with the mean and standard deviation of the original data. This means that the log normal distribution for this analysis needs to be entered as Log Normal with parameters (0.30654,0.099788). IF YOUR SIMULATION SOFTWARE PERMITS NEGATIVE PARAMETER VALUES USE THE FIRST VERSION. OTHERWISE USE THE VERSION WITH THE MEAN AND STANDARD DEVIATION. Please confirm with your simulation software user manual.

Figure 78 Chipotle ordering times and how DESI detects serial correlation.


CLASS II data histogram with distribution fit
File
Histogram of chipotlepay with best fit probability distribution.
Mean=0.52772 Standard Deviation=0.22509 Median=0.49167
Frequency
Time in minutes
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
Best fit
OK
CLASS II data autocorrelation diagnostics
Scatter plot of chipotlepay(t-1) versus chipotlepay(t)
Correlation= 0.35985 is not significant
chipotlepay(t)
chipotlepay(t-1)
Plot of chipotlepay. Blue horizontal line is the mean.
Calculated service time
observation number
Correlogram for chipotlepay
correlation
lag
Service time as a function of entry timestamp for chipotlepay.
Slope = -0.75877 is not significant and intercept =9.6914
Calculated service time
Timestamp of data entry
11.9
11.95
12
12.05
12.1
12.15
12.2
12.25

Log Normal distribution with parameters (-0.72599,0.41794) for N=26.
Please notice that some applications will require you to enter the log
normal distribution with the mean and standard deviation of the original
data. This means that the log normal distribution for this analysis needs
to be entered as Log Normal with parameters (0.52772,0.22509). IF YOUR
SIMULATION SOFTWARE PERMITS NEGATIVE PARAMETER VALUES USE THE FIRST
VERSION, OTHERWISE USE THE VERSION WITH THE MEAN AND STANDARD DEVIATION.
Please confirm with your simulation software user manual.

Figure 79 IID Chipotle payment times and autocorrelation diagnostics.

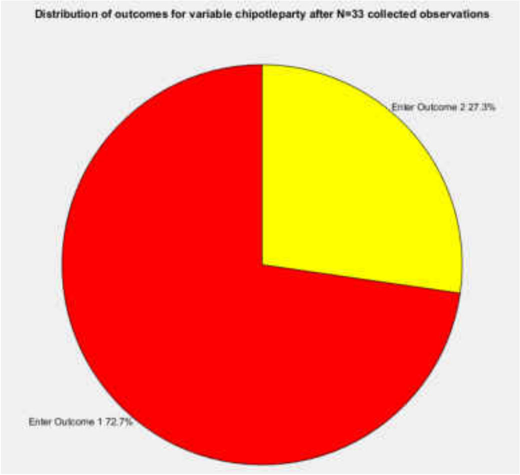


Figure 80 Distribution of customers entering Chipotle.

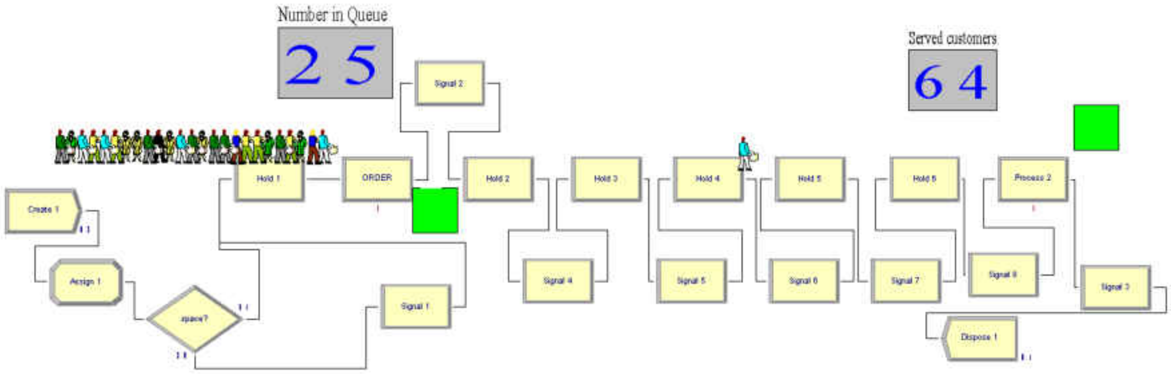


Figure 81 Chipotle Arena model

<u>Summary of Results</u>

The DES projects presented in this chapter show a significant positive impact from using the DESI interface in the percentage of DES projects' time devoted to input data management.

As presented in chapter three, the reduction in time devoted to input data management comes from eliminating post-collection data processing Λ_ρ, fulfilling condition (3.15), where $\Lambda_\rho \approx 0$. The projects presented in this chapter are not manufacturing applications of DES, since manufacturing applications of DES are already the focus of other documented solutions for DES automated data processing. The fact that no manufacturing project was included in this chapter does not imply that the proposed framework, instantiated by the DESI interface, cannot be used in manufacturing applications of DES.

Without DESI, the projects presented in this chapter would have been completed using methodology (A) and (B) solutions, namely MDCAP. More sophisticated solutions for DES input data automation would not have been adequate for these types of DES projects because the systems modeled in this chapter did not have previously available data, which is the most important requirement in order to use documented methodology (C) and (D) solutions.

The projects presented in this chapter are of reasonable complexity, but in reality DES models can get much more complex. **Figure 82** shows that as model complexity increased, the total hours devoted to input data management also increased, even with automation. Model complexity is measured as the sum of hours needed to develop the model logic plus the hours used in model creation within DES software. In **Figure 82**, model complexity is presented in the x-axis and hours spent in input data management are presented in the y-axis. Results show a significant positive linear correlation of 0.85 between model complexity and time devoted to input data management.

Even though model complexity implies more absolute time devoted to input data management, the percentage of time devoted to input data management decreased as model complexity increased. This means that the reduction in the percentage of DES projects devoted to input data management suggested in this chapter would not be compromised as model complexity increases; as a matter of fact, such reduction in percentage time devoted to input data management is reinforced as complexity increases.

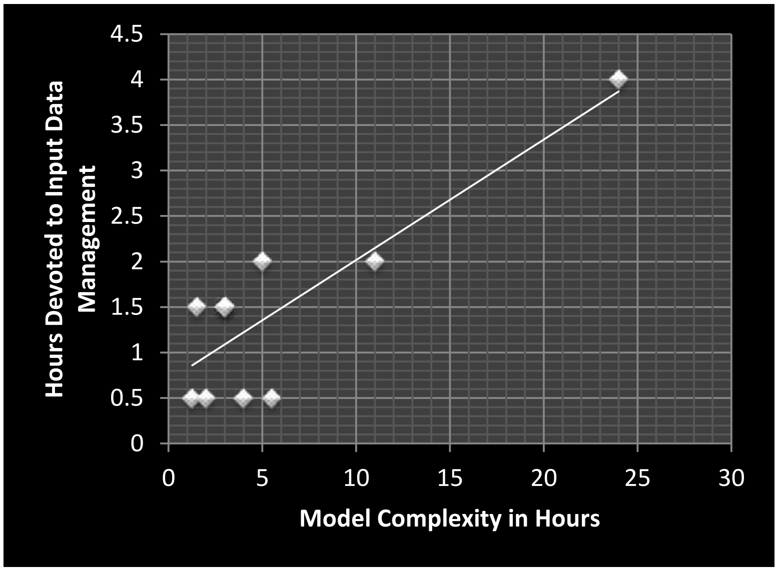


Figure 82 Input data management vs model complexity

CHAPTER SEVEN: FUTURE RESEARCH AND CONCLUSIONS

Place in DES Input Data Management Literature

This section reconciles the proposed framework with the literature about DES input data management. Skoogh and Johansson proposed the methodology for increased precision and rapidity in DES input data management shown in **Figure 83** [5]. This methodology emphasizes the identification and definition of relevant parameters, specification of requirements and availability of data as the initial steps in DES input data management [5]. Such steps can be regarded as equivalent to the concept of model logic presented in the proposed integrated framework. Using **Figure 83** as reference, the proposed framework presented in this research investigation permits data processing automation that starts in the "Create data sheet" step, continues through the right path, i.e. "gather non available data", and ends after the completion of the "Finish final documentation" step, since the proposed framework allows for streamlined validation the collected data. The aforementioned automation is possible due to the generalization of the data into three specific classes, as presented in the proposed integrated framework for DES input data management automation.

In summary, the documented steps from **Figure 83** indicate what needs to be done in DES input data management, whereas the proposed integrated framework presented in this research investigation proposes a methodology to get it done in an automated manner, regardless of the discipline or area of application.

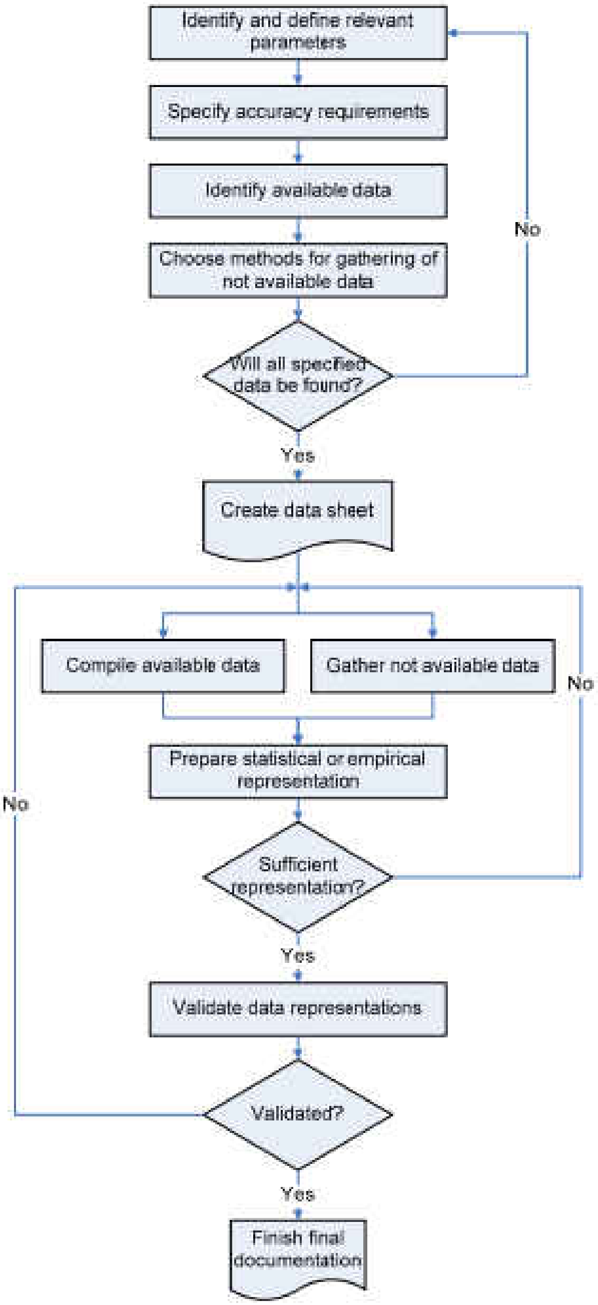

Identify and define relevant parameters
Specify accuracy requirements
Identify available data
Choose methods for gathering of not available data
Will all specified data be found?
No
Yes
Create data sheet
Compile available data
Gather not available data
No
Prepare statistical or empirical representation
No
Sufficient representation?
Yes
Validate data representations
Validated?
Yes
Finish final documentation

Figure 83 Methodology for increased precision and rapidity in input data management

Source: Skoogh, A. A., & Johansson, B. B. (2008). A methodology for input data management in discrete event simulation projects. 2008 Winter Simulation Conference Proceedings, 1727. doi:10.1109/WSC.2008.4736259

Future Research

In addition to speeding up data preparation for DES projects, the proposed framework, instantiated by the DESI interface, offers advantages to the users such as flexibility, data verifiability, lower implementation costs, ease of use by non-technical staff, real-time data visualization, real-time data fitting, real-time assumption checking, simulation-ready output, multimodal data fitting and relative frequencies of outcomes at decision points. User satisfaction from these convenient features needs to be tested in user experience evaluations, which are the subject of future research. Future research will also be focused on the financial impact of using DESI instead of using MDCAP or instead of using more complex methodology (C) and (D) solutions. The argument in favor of DESI as a less costly solution is the fact that it requires fewer resources and less trained technical staff.

Then research will be directed towards the usage of sensors in order to replace humans pressing the buttons in the DESI interface. Sensorial activation of DESI will not be feasible in all DES applications, but when applicable, sensors can help eliminate or avoid the unavoidable wait for events to happen η_ρ because sensor signaling will replace humans in the collection of the data. Based on the claims from chapter three, if DESI can respond to sensor signals, the time devoted to input data management will approach 0, namely $\beta_\rho \approx 0$. Such reasoning leads to the following,

Claim 7 Under the proposed integrated framework for automated DES data collection and processing, the condition where $\Lambda_\rho \approx 0$, $\eta_\rho \approx 0$, $\theta_\rho \approx 0$ and $\beta_\rho \approx 0$ for a DES project ρ is feasible if sensor signaling can replace human intervention in the collection of unavailable but collectable data.

Conclusion

This research investigation presented an integrated framework to automate data collection and processing for Discrete Event Simulation (DES) projects. Input data processing steps for DES projects include calculations, probability distribution fitting, visualization and validation. The motivation for selecting this topic is the fact that input data management for DES projects is a highly manual process. This is confirmed by the fact that manual input data methodologies (A) and (B) are employed by 80% of recently surveyed DES practitioners. Even though specialized DES software runs the simulations, it does not prepare the data to set up the simulations, hence there is the need for a solution that can automate input data management, while being usable in the different disciplines where DES modeling is appropriate.

The literature about DES input data management presents specific attempts to automate DES data preparation. Documented attempts are of the form suggested by methodology (C) and (D) solutions, where automation depends on the existence of data within internal corporate systems. Such sophisticated automation attempts present important drawbacks that prevent their proliferation. First, these sophisticated solutions require previously available data, which often do not exist. Second, their implementation can be challenging because of customized configuration requirements and the need for specialized staff. Third, these sophisticated solutions are presented only in manufacturing applications, which is one of the areas where DES modeling is appropriate.

The proposed integrated framework generalizes the data needed for DES projects into three classes. These classes are Class (I) data, where the ending timestamp of one event represents the beginning timestamp of the following event; Class (II) data, where each event has its own start and end timestamp; and Class (III) data, where counts of outcomes at a given decision point generate relative frequencies or probabilities. Such generalization permits automating the collection and processing for each one of these classes, given a model logic for the target system.

The proposed integrated framework provides the foundations to develop an innovative automated solution that combines key features from the four DES input data management methodologies (A), (B), (C) and (D). For example, the solution based on the proposed framework permits full verification and validation of the data at any time as would be expected in methodology (A) and (B) solutions, while achieving automatic processing, storage, extraction and updates of simulation-ready data, as would be expected in methodologies (C) and (D) solutions.

The automated, hybrid solution based on the proposed integrated framework is called DESI, which stands for Discrete Event Simulation Inputs. The DESI interface has three main sections, which are devoted to the collection and processing of Class (I), Class (II) and Class (III) data. The interface also features a section of optional parameters, which permits the selection of specific criteria in order to customize the user experience in the analysis of data for DES projects. DESI is activated by pressing buttons and it returns processing results in real-time. Simulation-ready results include probability distribution fitting, histograms, autocorrelation diagnostics, multimodal data partitions and relative frequencies in the form of pie charts. This research investigation presents two versions of DESI. These versions are DESI 1.0, which stores data in MySQL databases and DESI 2.0., which stores the data in text files.

This research investigation presented DES projects completed using the DESI interface. The completion of such projects showed that the proposed framework, instantiated by the DESI interface, can reduce the percentage time devoted to input data management from 31%, which is the average presented in the literature, down to 21%. Such reduction is statistically significant. The 21% obtained using DESI comes from the unavoidable wait for events to happen, as the implementation of DESI is very simple and post-collection data processing is fully automated based on the concepts of the proposed framework. Future research can be focused on the financial impact of the proposed framework as it requires fewer resources. Future research can also be focused on user experience evaluations of the DESI interface in order to improve its design and in the usage of sensors to replace humans pressing the buttons of the DESI interface.

A potential disadvantage of DESI is the fact that data must be collected by someone. In other words, the current versions of DESI need a person controlling the interface by pressing the buttons and checking the boxes in order to collect and process the data. Other documented DES input data management solutions obtain the data directly from internal data sources. Such reliance on internal data systems also has its drawbacks.

The challenges of relying on internal data sources have been documented in the literature. These challenges include complexity, high costs, lengthy implementation, detailed customization and lack of flexibility. Due to the complicated nature of input data management solutions that rely on internal data sources, DESI can be considered a convenient alternative in the automated processing of data for DES projects because it is independent of internal data sources. Hence, even though the lack of connectivity to internal data sources poses the need for someone to control the DESI interface, such independence also creates a significant strength. Non-reliance on internal data permits DESI to be usable in different areas where DES modeling is appropriate beyond manufacturing, advancing the state-of-the-art in automated DES input data management.

REFERENCES

[1] Bangsow, S. (2012). *Use cases of discrete event simulation: Appliance and research*. Berlin: Springer.

[2] Ward, Bryan, *A Framework for the Automation of Discrete-Event Simulation Experiments* (2011). Honor's Theses. Paper 42.

[3] A.M. Law, *Simulation Modeling and Analysis*, fourth ed., McGraw-Hill, New York, 2007.

[4] Kelton, W., Sadowski, R. P., & Sadowski, D. A. (2002). *Simulation with Arena* / W. David Kelton, Randall P. Sadowski, Deborah A. Sadowski. Boston : McGraw-Hill, c2002.

[5] Skoogh, A. A., & Johansson, B. B. (2008). A methodology for input data management in discrete event simulation projects. *2008 Winter Simulation Conference Proceedings*, 1727. doi:10.1109/WSC.2008.4736259

[6] Jahangirian, M., Eldabi, T., Naseer, A., Stergioulas, L. K., & Young, T. (2010). Simulation in manufacturing and business: A review. *European Journal Of Operational Research*, 203(1), 1-13. doi:10.1016/j.ejor.2009.06.004

[7] Robertson, N., & Perera, T. (2002). Automated data collection for simulation?. *Simulation Practice and Theory*, 9349-364. doi:10.1016/S0928-4869(01)00055-6

[8] Skoogh, A. A., Perera, T. T., & Johansson, B. B. (2012). Input data management in simulation - Industrial practices and future trends. *Simulation Modelling Practice and Theory,* 181. doi:10.1016/j.simpat.2012.07.009

[9] Robinson, S., & Bhatia, V. (1995). Secrets of successful simulation projects. *1995 Winter Simulation Conference Proceedings*, 61. doi:10.1145/224401.224424

[10] Robinson, S. S. (2004). Simulation: *The practice of model development and use*. John Wiley.

[11] Pentaho Business Analytics. (2013. Linux Journal, (235), 66.

[12] Methodology for rapid identification and collection of input data in the simulation of manufacturing systems. (2000). *Simulation Practice and Theory*, 7(7), 645.

[13] Moon, Y. B., & Phatak, D. (2005). Enhancing ERP system's functionality with discrete event simulation. *Industrial Management & Data Systems*, 105(9), 1206-1224. doi:10.1108/02635570510633266

[14] Skoogh, A., Johansson, B., & Stahre, J. (n.d). Automated input data management: evaluation of a concept for reduced time consumption in discrete event simulation. *Simulation-Transactions of the Society for Modeling and Simulation International*, 88(11), 1279-1293.

[15] Aufenanger, M., Blecken, A., & Laroque, C. (n.d). Design and implementation of an MDA interface for flexible data capturing. *Journal of Simulation*, 4(4), 232-241.

[16] Law, A. M. (2011). How the ExpertFit® distribution-fitting software can make your simulation models more valid. *2011 Winter Simulation Conference Proceedings*, 63. doi:10.1109/WSC.2011.6147740

[17] Swain, J. J. (2011). Simulation: back to the future: a brief history of discrete-event simulation and the state of simulation tools today. OR/MS Today, (5), 56.

[18] Corporation, Geer Mountain Software (2014). Stat::fit Distribution Fitting Software. Retrieved 23 April 2014 from Geer Mountain Software Corporation: http://www.geerms.com

[19] Law, A. M., & McComas, M. G. (1990). Secrets of successful simulation studies. Industrial Engineering, (5), 47.

[20] Skoogh A and Johansson B. Mapping of the time consumption during input data management activities. Simulation News Europe 01/2009; 19(2):39-46. 2009; 19: 39–46.

[21] Bengtsson, N., Shao, G., Johansson, B., Lee, Y., Leong, S., Skoogh, A., & Mclean, C. (2009). Input data management methodology for discrete event simulation. Winter Simulation Conference, 1335.

[22] Preece, J., Rogers, Y., & Sharp, H. (2002). *Interaction design : beyond human-computer interaction* / [Jennifer] Preece, [Yvonne] Rogers, [Helen] Sharp. New York, NY : J. Wiley & Sons, c2002.

[23] Sigmon, K., & Davis, T. A. (2002). *Matlab Primer*. Boca Raton: Chapman & Hall/CRC.

[24] Johansson, M. M., Johansson, B. B., Skoogh, A. A., Swee, L., Riddick, F. F., Lee, Y. T., & ... Klingstam, P. P. (2007). A test implementation of the core manufacturing simulation data specification. *2007 Winter Simulation Conference Proceedings*, 1673. doi:10.1109/WSC.2007.4419789

[25] Johansson M, Johansson B, Leong SK, et al. A real world pilot implementation of the core manufacturing simulation data model. *Proceedings of the summer computer simulation conference*, Edinburgh, Scotland, 2008.

[26] Shore, H., & A'wad, F. (2010). Statistical Comparison of the Goodness of Fit Delivered by Five Families of Distributions Used in Distribution Fitting. *Statistics: Theory & Methods*, 39(10), 1707-1728. doi:10.1080/03610920902887707

[27] Johansson, B., Skoogh, A., Mani, M., & Leong, S. (2009). Discrete event simulation to generate requirements specification for sustainable manufacturing systems design. *Proceedings of the 9Th Workshop: Performance Metrics for Intelligent Systems*, 38. doi:10.1145/1865909.1865918

[28]. Kibira D and Leong SK. Test of core manufacturing simulation data specification in automotive assembly. *Proceedings of the Simulation Interoperability Standards Organization (SISO) and Society for Modeling and Simulation (SCS) international European multi conference*, Orlando, FL, 2010.

[29] Portnaya, Irin. (2004). An approach to automating data collection for simulation. (Master Thesis). Retrieved from University of Central Florida Library. (Accession Number ucfl.023120114)

[30] Kardos, C., Popovics, G., Kádár, B., & Monostori, L. (2013). Methodology and Data-structure for a Uniform System's Specification in Simulation Projects. Procedia CIRP, 7455. doi:10.1016/j.procir.2013.06.015

[31] Li, Y., & Joshi, K. D. (2012). Data Cleansing Decisions: Insights from Discrete-Event Simulations of Firm Resources and Data Quality. Journal Of Organizational Computing & Electronic Commerce, 22(4), 361-393. doi:10.1080/10919392.2012.723588

[32] Davenport, T. H., & Harris, J. G. (2007). *Competing on analytics: The new science of winning*. Boston, Mass: Harvard Business School Press.

[33] Trybula, W. 1994. Building simulation models without data. *1994 IEEE International Conference on Systems, Man, and Cybernetics. Humans, Information and Technology*, 1:209-214. IEEE

[34] J. L. Romeu,. Kolmogorov-Smirnov: A Goodness-of-Fit Test for Small Samples. RAC START, volume 10, number 6, 2003.

[35] Galitz, W. O. (2002). *The Essential Guide to User Interface Design : An Introduction to GUI Design Principles and Techniques*. New York: Wiley Computer Pub.

[36] Johnson, J. D. (2010). Designing with the Mind in Mind [electronic resource] : Simple Guide to Understanding User Interface Design Rules. Burlington : Elsevier, 2010.

[37] Coulson, L. (2009*). Matlab Programming*. Chandni Chowk, Delhi: Global Media.

[38] MySQL, A. (2006). *MySQL Administrator's Guide and Language Reference*. Indianapolis, Ind: MySQL.

[39] Horn, J. D., & Grey, M. (2004). *MySQL : essential skills* / John Horn, Michael Grey. Emeryville, Calif. : McGraw-Hill/Osborne, c2004.

[40] Bell, C. A. (2012). *Expert MySQL* [electronic resource] / Charles Bell. Berkeley, CA : Apress ; New York : Distributed to the book trade worldwide by Springer, c2012.

[41] Vaswani, V. (2004). *MySQL : the complete reference* / Vikram Vaswani. New York : McGraw-Hill/Osborne, c2004.

[42] Everitt, B., & Skrondal, A. (2010). *The Cambridge Dictionary of Statistics*. Cambridge: Cambridge University Press.

[43] Neter, J., Wasserman, W., & Kutner, M. H. (1990). *Applied linear statistical models: Regression, analysis of variance, and experimental designs*. Homewood, IL: Irwin.

[44] Parasuraman, R., & Sheridan, T. B. (2000). *A Model for Types and Levels of Human Interaction with Automation*. IEEE Transactions On Systems, Man & Cybernetics: Part A, 30(3), 286.