Proceedings of the 2016 Winter Simulation Conference
T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, eds.

LANGUAGE-AGNOSTIC SIMULATION MODEL MANAGEMENT PLATFORM

Genevieve Brown
Taylor Curro
Ryan Gentry
Thomas Hoffman
Timothy Lortz
Joseph Murrey
Joshua Peters
Scott Rachlinski
Eric Zatcoff

Booz Allen Hamilton
308 Sentinel Drive
Annapolis Junction, MD 20701, USA

ABSTRACT

Simulation models are expensive to design and run. Commercial modeling software made for specific industries are costly and do not allow for compatibility across languages. Free and open source languages allow for more control in model creation. However, models written using these languages require more time to develop and interpret. We present a platform compatible with open source languages that runs models in Monte Carlo and Design of Experiments sampling with the ability to randomly distribute inputs and interpret outputs. Supported languages include Java, Python, and R. The platform utilizes a parameter input configuration syntax which defines how model inputs are distributed, allowing for both deterministic and stochastic model creation. Models outputs are then interpreted by the platform using built-in customizable tools.

1 INTRODUCTION

Modeling and simulation can be performed using a number of different software packages and languages. The process can be broken down into four steps: 1. Research and Develop Model 2. Run Model 3. Analyze Results 4. Repeat. This program seeks to tie this entire process into one solution. The modeler is able to edit, execute, and visualize their model's output without the need for any commercial or external software.

2 LANGUAGE-AGNOSTIC FUNCTIONALITY

Connecting languages into a single interface eliminates the need for additional software and allows users to develop models in different varieties. The program's multi-language capability is facilitated by implementing the third-party service RServe for R and by automated command line code execution for Python and Java. Models are run in the background by these protocols as instructed by the main program.

Compatibility issues between models and the program are resolved by carefully coding how models receive inputs and return outputs. Each language follows a similar statement for receiving and sending variables to and from the program. Once this adaptation is made, the model is fully compatible with the program.

3 MODEL INPUT CAPABILITIES

Traditionally, input variables are defined above the model in the same file. In this program, models and their inputs are abstracted from one another to allow for more control over how inputs are defined. The configuration file provides instructions on how to handle each input. The program currently supports sampling from 12 different distributions and input types. Doing this allows for the program to come up with generic input statements which are compatible with any supported language.

4 RUN METHODS: MONTE CARLO AND DESIGN OF EXPERIMENTS

The run methods supported by the program include Monte Carlo simulation and Design of Experiments. In Monte Carlo simulation, the user must define the number of iterations. Upon loading the model file and the input configuration file, the program has all of the instructions it needs to run the simulation. For each iteration of Monte Carlo, a new random sample of input parameters is taken and the model output is captured. After the simulation is finished, the modeler may then proceed to analysis.

In Design of Experiments, the configuration and model files are loaded and the user defines full or fractional factorial mode. The user can also specify the design matrix through a graphical user interface. After running, main effects and interaction plots are displayed, allowing the modeler to see how variables impact the output.

5 ANALYTICS

Built-in analytics were created to allow modelers to quickly draw insights on outputs. Although the program provides these, it is possible to change or add new custom analyses to the program by editing the supporting files.