

DOING SIMULATION FOR A LIVING

Mohamed Sam Fayez
Fabio Zavagnini

Productivity Apex, Inc.
3505 Lake Lynda Dr., Suite 206
Orlando, FL 32817, USA

Mansooreh Mollaghasemi

University of Central Florida
4000 Central Florida Blvd.
Orlando, FL 32816, USA

ABSTRACT

For more than half a century, simulation has proven to be a successful companion to decision makers and analysts around the world, with several packages being successfully used by thousands of modelers, our team being one of them. Despite great popularity and success, however, there have been issues and shortcomings that may threaten the accuracy and efficiency of the model as a decision-making tool. In this light, working with simulation for a living requires special requirements in order to ensure success. As such, there are several experience-based practices that can reverse the negative effect of these issues and even position simulation at a higher rank among decision-making tools. This shall be a summary of the issues and shortcomings faced by our simulation team throughout the past ten years (2001 – 2011) followed by a presentation of the team's best practices, which ultimately lead to a more efficient and streamlined simulation management tool and methodology.

1 INTRODUCTION

Simulation modeling methodology guides the modeler through a series of steps to develop and deliver a credible decision support solution. These steps start by identifying and defining the problem, developing a conceptual model, and identifying data requirements. The modeler must build the model, populate it with data, verify and validate the model, run and analyze the newly-modeled system, build and run what-if scenarios, and sometimes build a graphical user interface for the model. When the model is ready for decision making, the modeler delivers it to the customer or provides analysis reports from the simulation scenarios. In both cases, the modeler will always support the developed model via updates, more scenarios, and more analysis reports. There are several commercial simulation software products that vary in features and options in order to facilitate the simulation methodology, but very few empower the modeler's or the user's post-model development. The general simulation methodology is shown in Figure 1.

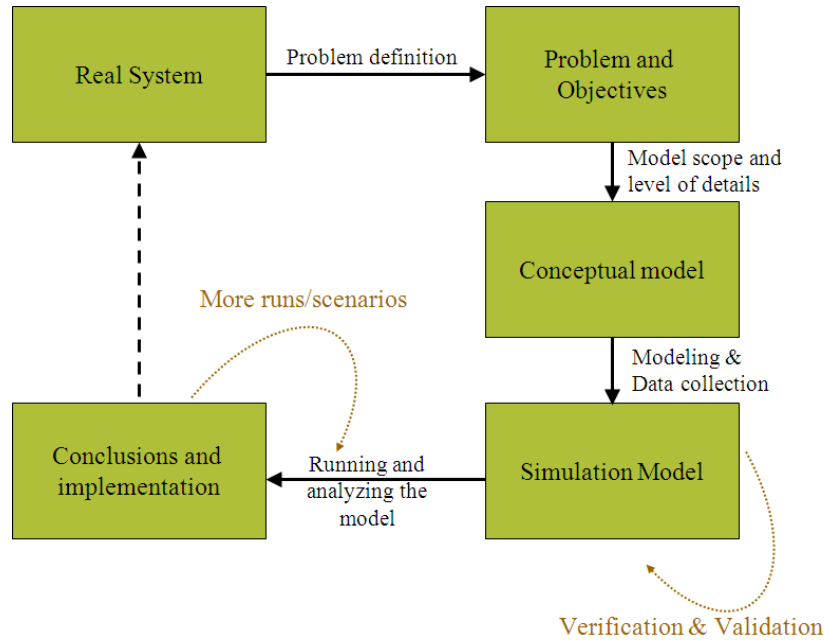


Figure 1: General Simulation Methodology

Our team extended the general simulation methodology to capture the steps practiced after the model's development, including the final delivery to the customer. Figure 2 shows the extended methodology.

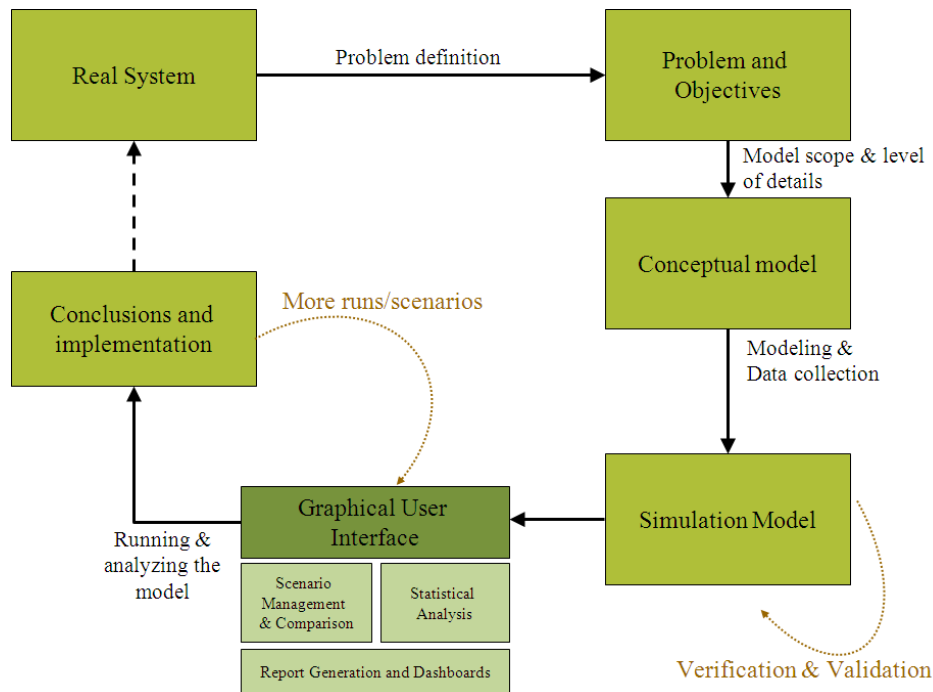


Figure 2: Extended Simulation Methodology

Our modeling team developed simulation models using the leading simulation software in the market, strictly following the simulation methodology described in Figure 2 (above).

2 ISSUES AND SHORTCOMINGS

A modeler usually builds the simulation model using one of the simulation software programs available in the market such as Arena, Promodel, Simio, FlexSim, Anylogic, and so on. The modeler uses the simulation software to execute the modeling and analysis steps mentioned above. The time to build the model varies between modelers as well as their modeling techniques used. In general, most modelers gain modeling experience fast; and having multiple modelers working closely greatly affects their learning curve and the efficiency of their modeling techniques. In this section, the issues and shortcomings discussed takes into consideration only experienced modelers in order not to bias the findings with issues due to lack of modeling experience.

After building the model, the modeler will interact with the model through the simulation software used, which usually requires time, effort, and precision from the modeler to be able to execute the remaining steps in the methodology before delivering the model or start running simulation scenarios and analysis tasks. For example, if the modeler would like to make any changes in the values of the input parameters, all the model blocks to make any necessary changes must be edited. This task becomes challenging for more complex models or when the number of changes increases. Some modelers have opted for creating custom-built Excel sheets to manage input and output; however, this still requires considerable set-up time, usually allowing the user to run only one scenario at a time and demanding programming skills for more advanced functionalities. The modeler's issues and shortcomings have direct and indirect impacts on time, cost, quality, and customer service. These include:

- The modeler usually interacts directly with the model in the used simulation software one module at a time to make any changes in the model logic or data.
- The modeler usually interacts directly with the simulation software over several iterations during the verification, validation, analysis, and running what-if scenarios.
- The modeler generates only one output report for each simulation run. This report will be overwritten after each simulation run. Thus, the modeler saves individual reports per run or copies the output of interest to a third party software.
- The modeler could create custom-built Excel sheets and link it to the model. This sheet will act as an interface which hosts model input and simulation run output. However, the output data will be overwritten by the last simulation run. Thus, the modeler should save a different version of the Excel sheet per simulation run or copy the sheet data to another sheet. Also, implementing this solution requires extra modeling knowledge and effort; for example, with Arena®, the modeler will need to insert special read/write modules in the model to be able to read from and write to Excel, in some cases involving programming in VBA to directly link model variables to spreadsheet cells.
- The modeler must build custom reports or charts to present the model output, sometimes using a third party software. In similar manner, the modeler will need to further build the report to compare multiple simulation runs and what-if scenarios. Also, the report must be re-built, sometimes manually, to accommodate any changes in model output scope (e.g. adding a new output parameter).
- The modeler, if required, must run statistical tests (e.g. t-tests, precision tests, correlation analyses) on model output or must compare scenarios, requiring completion of these tests using third party tools and necessitating custom reports on the findings.
- Any modeler other than the original model developer needs the time to be familiar with the model before making any changes or even running simulation scenarios and generating analysis reports.
- The modelers are interrupted by customers each time they need to run a different scenarios or conduct additional analysis.
- The model stakeholders will not be able to use the model or run scenarios without modeler support. The modeler must support the stakeholders and the model during its whole life cycle.

Similarly, users and decision makers of the model will face some issues, shortcomings, and limitations in using the model, including:

- The user relies completely on the modeler to run simulation scenarios or to make any changes to the model. This may delay utilizing the analysis from the model for decision making in a timely manner due to resource availability and priorities.
- The user receives a separate report for each simulation run. This will demand time for compiling each report separately and to manually build comparison reports if several scenarios are to be compared.
- The user manually creates comparison reports to be able to compare simulation scenarios side-by-side or relies on the modeler to build such comparisons, but at the cost of further stretching the analysis cycle time.
- The user is required to build specific spreadsheets to perform statistical tests in order to compare the output (e.g. t-test).
- It is difficult to focus on specific output parameters unless they are exported separately to a predefined Excel sheet as variables.

All these issues require significant time and effort from the modelers during and after the model development. Support for models becomes a significant time drain and greatly impacts new projects. Often, users and decision makers are delayed in decision making due to modeler's lead time. In addition, scenarios are often kept to a minimum due to time restraints and cost when the ideal situation would provide more scenarios and variables, achieving maximum results from simulation modeling in a shorter period of time. In reference to these issues, simulation modeling and analysis cycle time becomes significantly high in practice, since these limitations make it time-demanding and a tool that requires too much effort and specific expertise to be executed successfully. In order for simulation modeling and analysis to be easier to implement and operate, these issues and shortcomings must be minimized or ultimately eliminated.

3 OVERCOMING ISSUES AND SHORTCOMINGS, ONE MODEL AT A TIME

Simulation modeling and analysis is the core business of our team. Thus, the issues and shortcomings were directly impacting our productivity, cost efficiency, and customer service. The management and key team members have been continuously working to streamline the applied simulation methodology and deliver better simulation experience to customers. The main focus of this section is to present the team's practices and progress within the past decade (2001 – 2011) to overcome the aforementioned obstacles. All the projects presented below are real-life simulation projects that have delivered valid models and/or analysis studies. The focus shall be models developed using Arena®.

One of the first simulation models our team developed was a macro-level simulation model for the Space Shuttle operations held at the Kennedy Space Center (KSC). This model was developed to answer questions regarding processing times and the number of orbiters, processing facilities, vehicle assembly buildings, launch pads, etc., on the expected flight rate and the utilization of each facility. The model was developed, and delivered to the customer. The development team was interacting with the model through Arena. All the required changes in the model data input was done through editing the Arena modules in the model. The model output was reviewed through Arena crystal reports one scenario at a time. The customer was not able to use the model due to lack of experience in using Arena and requested comparison reports between some simulation scenarios from the development team. The team executed the simulation scenarios and compiled the comparison reports.

In the second phase of this project, the model was extended and the team decided to develop an Excel sheet to be used in conjunction with the model. The sheet was used to change model input to ease the task of running multiple scenarios. Also, the model was programmed to write specific model output to the Excel sheet. The model developers, analysts, and customers all saw the use of the Excel sheet with the model as a step forward in usability and analysis efficiency. However, using the Excel spreadsheet required

extra effort, albeit in different direction. The model had to be modified by adding read/write modules, defining files, adding attributes and variables, defining the spreadsheet range, adding expressions to allow multiple module updates, and adding VBA code to communicate with the spreadsheet. Also, the Excel spreadsheet had to be enriched with some macros, adding some VBA code, and a unique name definition of the cell range from which data will be read by the model or to which data will be written during/after the simulation run. At the file management level, the model communicates with only one file with a specific name, so the model will write to the same file every run. Thus, the user had to rename and save a version of the excel file for each simulation scenario. Despite all extra work, the modeling and analysis experience was much better than using Arena alone.

The Excel sheet approach has been used after the success with the shuttle model for several projects, and in each project, the experience was enhanced by adding more macros and VBA code to automate some of the routine tasks, such as saving the Excel file on a different name after each run, resetting the excel spreadsheet before running a new scenario, etc. Also, the modeling time with this approach was shortened such as through use of a generic VBA code and then reusing it between projects. In general, the Excel approach had success, but it in fact was very limited and bothersome. The team decided to find a better approach to further enhance the simulation methodology and the experience of the stakeholders.

Building on the shuttle experience, the team developed a Generic Environment for Modeling Future Launch Operations (GEM-FLO). The model was developed, for the first time, with a standalone graphical user interface, shown in Figure 3.

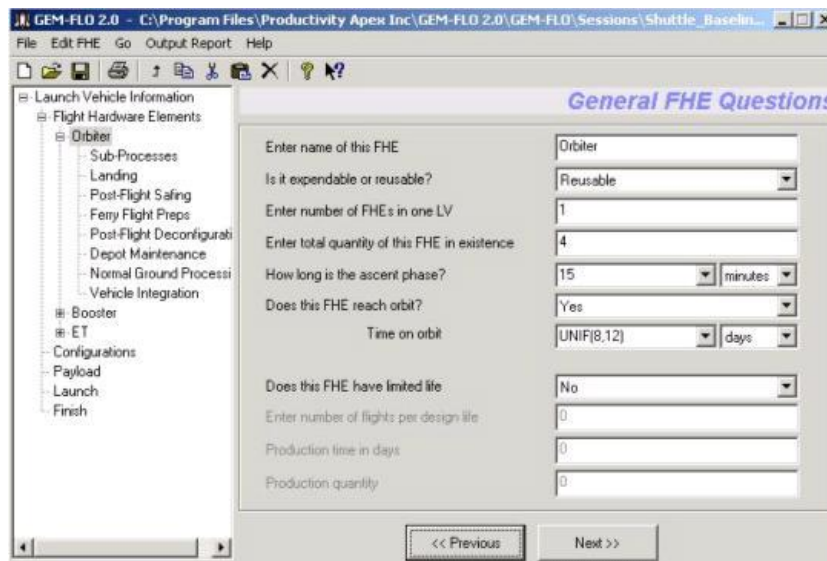


Figure 3: GEM-FLO Interface

The user interface was developed for use of the analyst and/or the user to customize the generic model and run simulation scenarios. The user interface allowed the users to enter information into the generic simulation model using domain-specific terminology with only basic knowledge of the simulation model. Subsequently, this same terminology was used in displaying the output results from running the model.

As the user goes through the process of entering data, a navigation tree is also constructed that reflects the data entered by the user. The tree serves as an organizational view of the data entered and provides easy access to different parts of the model.

After the user enters all model data in the interface, the user is presented with the "run parameters" window to specify the simulation run length, number of replications, and warm-up period. The user then runs the simulation models and generates the output report. The user can then save a session file with the

information entered. This saved session can be recalled to re-run a scenario or to edit and run a different scenario. The graphical user interface experience was a much better experience than the Excel spreadsheet practiced in earlier projects, according to common project stakeholders. The modelers and analysts were able to run simulation scenarios and generate output reports easier and faster than with the Excel spreadsheet. The customers, including those with no simulation or model knowledge, were able to use the interface to update model information, run simulations, and review output. At the file management level, the interface enabled the users to save session files and output reports. Developing the user interface required a parallel execution of a software development project that added software engineers to the project team and exhausted part of the project's budget. Despite all the extra work and programming, the modeling and analysis experience was much better than using Arena and Excel alone. The custom interface contributed in overcoming some of the issues and shortcomings mentioned earlier in Section 2.

The approach of building a custom graphical user interface has been used after the success of GEM-FLO model for several other models. The interfaces were developed in less time because of the experience gained by the team as well as the reusability of certain parts of the developed software code. Despite the success of the custom interface, as it was originally designed during GEM-FLO project; the team decided to redesign the interface to further empower the analysts and users.

The team was selected to model the terminal at Orlando International Airport to deliver an end-to-end model for passenger flow through the terminal. The airport authority requested that the model be used as a strategic tool as well as an operational decision support tool. The team developed the model and developed a new interface to be used with the model. The interface, named AirSim, was a merge between the design of the GEM-FLO interface and the spreadsheet interfaces developed for the shuttle model. The interface developed enabled modelers, analysts, and the customer to easily manage scenarios, view multiple scenario inputs and outputs side by side in a manner similar to a spreadsheet, setup a hierarchy of input categories, and view output reports with model output including average, half width, min, max, and number of replication for each output parameter. The interface also enabled the users to select the model to use and setup simulation run parameters. A screenshot of AirSim is shown in Figure 4. AirSim interface is composed of two frames. The left frame is the model browser, and the right frame has the attributes and the corresponding data from each operational scenario. Once the different scenarios have been developed, the users can populate and run multiple scenarios with a single press of a button. Output reporting is enhanced over the GEM-FLO interface as AirSim output reports can be customized only to show the output for particular functional area or for the whole airport, eliminating the time-consuming task of searching through the myriad of all statistics provided.

AirSim empowered the modelers, the analysts, and the final user of the model. It allowed customers with no simulation or model knowledge to do simulation modeling and analysis using AirSim. The success of AirSim interface positioned such practice as an excellent solution in overcoming many of the issues and shortcomings discussed previously. The team decided to develop similar interfaces for future simulation projects, taking into consideration some of the feedback and recommendations from AirSim stakeholders.

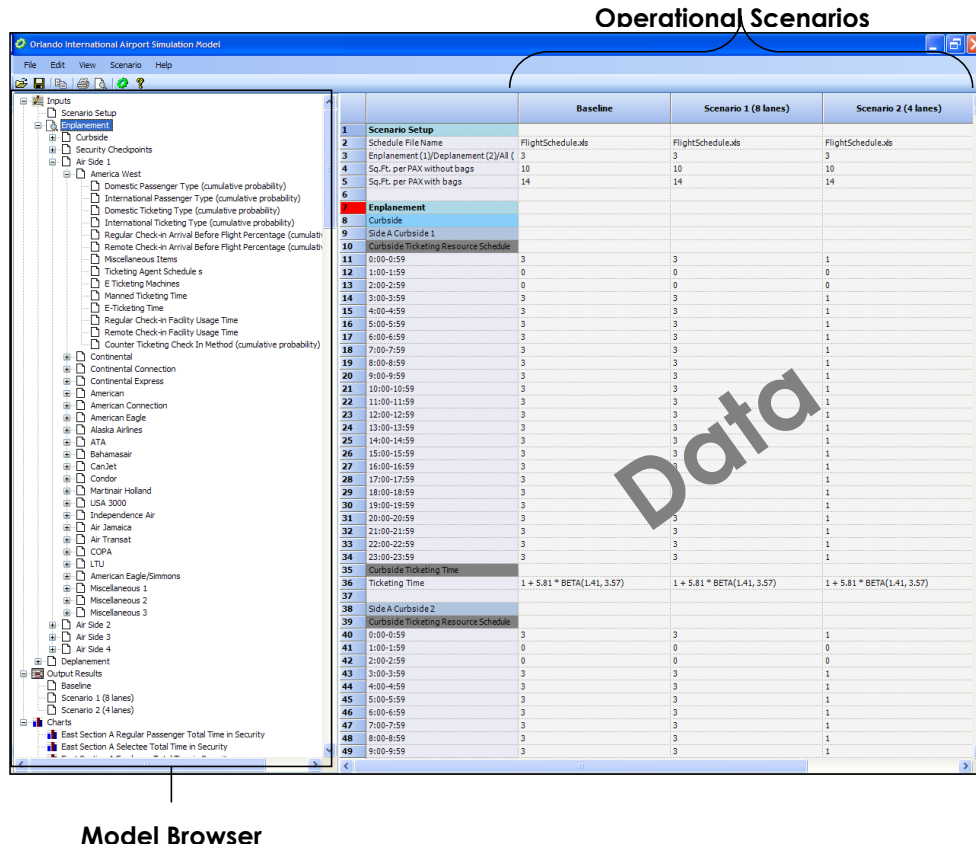


Figure 4: AirSim Interface

4 A BETTER SIMULATION EXPERIENCE, THE FRUIT OF THE YEARS

Building on the success of AirSim, the team decided to invest and develop a similar yet generic interface that can be customized for any simulation model in future projects. A core team was assembled that consists of simulation experts, modelers, analysts, and software engineers to define the requirements of the generic interface. The team agreed that the main objective of the new design was to deliver a better simulation experience that would limit or ultimately reverse the issues and shortcomings mentioned in Section 3. In that effort, the team conducted several brainstorming sessions to identify features required. The sessions included modelers, analysts, users, customers, and decision makers. The team also gathered the experiences and best practices of past interfaces, the lessons learned from simulation projects, and customer feedback. The following is the list of requirements along with their benefits that the team formulated for the interface design:

1. Have a shorter cycle time to build simulation models and run analysis scenarios. This allows the modeler to deliver results in much less time.
2. Have reduced testing time of the model, which would enable more in-depth verification. This will enable faster delivery of higher quality models.
3. Have reduced validation time of the model, which will enable more thorough validation testing and easier validation reporting to customers. This will enable delivering higher credibility models in less time.
4. Have an easier and faster ability to provide additional simulation scenarios to customers. This will enable higher customer service levels and better cost efficiency.

5. Have the ability to enable the model users and customers to run scenarios without relying on the modelers to do so. This will significantly reduce support time and empowers the users to run the required analysis and simulation scenarios. It will also enhance the practicality of simulation as a decision support tool.
6. Have the ability to enable analysts and users to select multiple scenarios at the same time to populate, run, and bring the results back to the tool.
7. Have less risk in making an error when changing model data or running new scenarios. This will enable delivering higher quality models and analysis reports.
8. Have the ability to allow any modeler to work on the model post-development, to run simulation scenarios, and to generate analysis reports. This will enable modelers or analysts to work on any model, thus eliminating the constraint that each modeler runs only the model developed by that specific modeler.
9. Have no need for building custom user interfaces for the simulation models. This will also eliminate the requirement for software engineers to develop such interfaces, and it will allow cheaper models to be delivered faster, empowering the modelers throughout the model's life cycle.
10. Have the ability to enable the modeler to select the model version to connect to the interface, to build the input hierarchy to easily browse through input parameters, and to select the input parameters to view in the interface.
11. Have the ability to easily generate reports that compare simulation scenarios side by side. This will enable higher quality analyses through better analysis reports and scenario comparisons. This will also eliminate the need to extract the simulation run data and build custom analysis spreadsheets in Excel or similar software, which will lead to faster delivery of scenario analysis results.
12. Have the ability to focus on specific output parameters. This will enable the modelers to customize simulation output and analysis to focus on the user's requirements.
13. Have the ability to rename input or output parameters to better match domain language. This will minimize the post-delivery requirements of making changes to the language or terms used in the models and output reports.
14. Have the ability to easily run statistical tests to compare output parameters across simulation scenarios. This will eliminate the need to build custom spreadsheets and mathematical equations to run statistical tests, which leads to faster and cheaper analysis delivery.
15. Have the ability to provide advice on simulation run parameters such as number of replications, run length, etc. This will enable users with no simulation experience to run sound simulation experiments.
16. Have the ability between simulation scenarios to highlight the changes in the inputs that lead to changes in the output .
17. Have the ability to view or hide sections of the input parameters and be able to move their locations as well as between input sections.
18. Have the ability to customize the output parameters by viewing and hiding output categories.
19. Have the ability to build charts of different types and formats to view simulation output and compare simulation scenarios graphically.
20. Have the ability to duplicate a scenario to alter and create a new scenario easily.
21. Have the ability to view or hide scenarios and change the order of the scenarios.

The core team designed the generic interface using the requirement list discussed above. The team called the interface *Ariana*, the first version being developed in 2007, fulfilling requirements 1 through 13. *Ariana* was successfully used in several projects and was ranked by modelers, analysts, and repetitive customers as the best simulation interface developed by the team. Building on the success of *Ariana*, the team later updated the interface in 2010. In the new version, several of the existing features were enhanced in terms of usability, and new features were added, the most important of these fulfilling requirements 14 through 21. *Ariana* successfully overcome the above-mentioned shortcomings and issues

in the simulation methodology as well as issues discovered in previous interfaces developed. The current version of *Ariana* can be used with any Arena® simulation model without the need to interact with the simulation model directly. *Ariana* even starts Arena® in the background and runs selected scenarios, specifically resulting in the following results being achieved:

- Faster delivery of analysis results, reports, and charts.
- Faster delivery of valid simulation models by an average of 30%.
- Reduction in verification and validation time by an average of 20%.
- Delivery of higher quality models and tools.
- Significant reduction in model support time by an average of 60%.
- Ability to run scenarios and generate output reports without relying on modelers.
- Reduced possibility of errors when making changes to the model, 90-100% reduction.
- Ability for any modeler or user to run scenarios and generate analysis reports.
- Ability to quickly create and run scenarios while also highlighting the data changes between scenarios as well as a selected baseline scenario.
- Ability to compare simulation scenarios side by side.
- Ability to run simulation scenarios in the background, thus able to run hundreds of scenarios without being in front of the computer.
- Ability to run statistical tests between scenarios: t-tests and highlighting statistically significant differences.
- Ability to focus on specific output parameters.
- Ability to quickly rename input or output parameters.

Ariana has been used by two types of users: modeler/analysts, referred to as *modelers*; and customers/users that not necessarily have simulation or underlying model knowledge, referred to as *users*. The *modelers* build, verify, validate, and deliver simulation models; often, they also do the analyses by using the model, creating simulation scenarios, comparing scenarios, running statistics tests, and delivering reports. The *users*, on the other hand, perform analyses through building what-if scenarios, running the simulation models, comparing/analyzing outputs, and generating analysis reports; but the *user* may not be the developer of the simulation model. *Ariana* was developed by utilizing more than ten years of expertise and intensive development of simulation models and simulation-based decision support solutions. *Ariana* was developed to fulfill the needs of modelers and customers and to streamline simulation modeling and analysis processes. A screenshot of *Ariana* 2.0 interface with description of the different interface sections is shown in Figure 5.

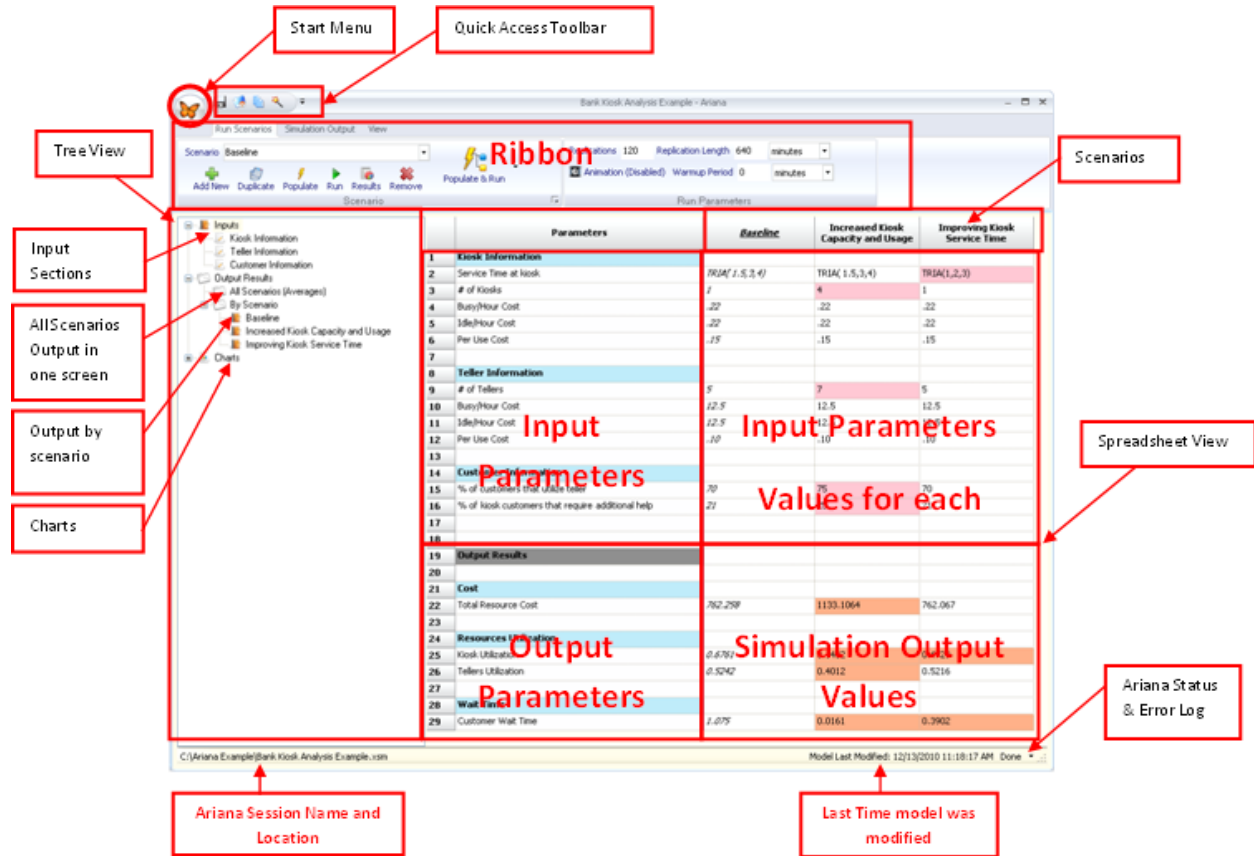


Figure 5: Ariana 2.0 Interface

REFERENCES

- Banks, J., J. S. Carson, B. L. Nelson, and D. M. Nicol. 2000. *Discrete-Event System Simulation*. 3rd ed. Upper Saddle River, New Jersey: Prentice-Hall, Inc.
- Bansal, S. 2002. "Promise and Problems of Simulation Technology in SCM Domain." In *Proceedings of the 2002 Winter Simulation Conference*, edited by E. Yücesan, C. H. Chen, J. L. Snowdon, and J. M. Charnes, 1831 – 1837. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc..
- Fayez, MS, Kaylani, A., Cope, D, Mollaghasemi, M., Zapata, E, Callinan, M. 2006. "Earth to orbit logistics and supply chain modeling and simulation for NASA exploration systems." In *Proceedings of the 2006 Winter Simulation Conference*, edited by L. R. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, 1462 - 1469. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc..
- Fayez, MS, Kaylani, A., Cope, D, Mollaghasemi, M. 2007. "Supply chain simulation modeling made easy: an innovative approach." In *Proceedings of the 2007 Winter Simulation Conference*, edited by S. G. Henderson, B. Biller, M.-H Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 1887 - 1896. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

- Fayez, MS, Kaylani, A., Cope, D, Rychlik, N, Mollaghasemi, M. 2008. "Managing Airport Operations Using Simulation" *Journal of Simulation*. 2:41-52
- Fayez, MS. Steele, M, Cates, G., Miranda, D., Mollaghasemi, M, Trocine, L. 2010. "NASA Constellation Program End-to-End Discrete Event Simulation System Analysis" *SpaceOps 2010 Conference AIAA 2010-1996*, April 2010.
- Kaylani, A., Mollaghasemi, M., Cope, D, Fayez, MS, Rabadi, G, Steele, M. 2007. "A Generic Environment for Modeling Future Launch Operations - GEM-FLO: A Success Story in Generic Modeling" *Journal of the Operational Research Society* 59, 1312-1320.
- Mollaghasemi, M., M. Steele, G Rabadi, and G. Cates. 2002. A Generic Environment for Modeling Future Launch Operations (GEM-FLO). *World Automation Congress*, June 2002, Orlando, Florida.
- Nayani, N., and M. Mollaghasemi. 1998. "Validation and Verification of the Simulation Model of a Photolithography Process in Semiconductor Manufacturing," In *Proceedings of the 1998 Winter Simulation Conference*, edited by D. J. Medeiros, E. F. Watson, J. S. Carson, and M. S. Manivannan, 1017 - 1022. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc..

AUTHOR BIOGRAPHIES

MOHAMED SAM FAYEZ, Ph.D., possesses extensive professional experience and educational credentials in the area of Industrial Engineering, Simulation, and Supply chain management. He is a Senior Industrial Engineer and the director of Supply Chain at PAI. Dr. Fayez has more than 15 years of diversified experience in management, engineering consulting, and research & development. Dr. Fayez holds a Ph.D. in Industrial Engineering from the University of Central Florida and has directed projects and initiatives for government agencies and private organizations. Dr. Fayez directed NASA's Exploration Supply Chain Modeling and Analysis project. He also led two separate efforts that involved simulation modeling and analysis for terminal operations at Orlando International Airport (OIA/MCO) while also working at OIA's Procurement Department. His email address is sfayez@productivityapex.com.

MANSOOREH MOLLAGHASEMI, Ph.D., is a tenured Associate Professor in the Department of Industrial Engineering and Management Systems at the University of Central Florida. Her research interest is in simulation modeling and analysis, multi-criteria optimization, and supply chain management. She is a member of IIE and INFORMS and was elected to serve on the board of NCS (National Center of Simulation). Her email address is mollagha@mail.ucf.edu.

FABIO ZAVAGINI is an Industrial Engineer with solid experience in the areas of statistics, simulation, GUIs, production, and inventory control, not to mention reliability engineering. Mr. Zavagnini has broad hands-on experience in the manufacturing, construction, and transportation industry. Over the past few years, he worked in numerous projects for commercial companies as well as government agencies. Mr. Zavagnini obtained his Master's degree in Engineering Management at the University of Central Florida. His email address is fabio@productivityapex.com.