Proceedings of the 2024 Winter Simulation Conference
H. Lam, E. Azar, D. Batur, S. Gao, W. Xie, S. R. Hunter, and M. D. Rossetti, eds.

CRACK THE CODE: EXPERT STRATEGIES FOR VERIFICATION AND DEBUGGING IN SIMIO

Madison Evans[1,2] and Jeffrey S. Smith[1]

[1]Simio, Sewickley, PA, USA
[2]Industrial and Systems Engineering, Auburn University, Auburn, AL, USA

ABSTRACT

In the age of Digital Twins and Artificial Intelligence, data is everywhere, and the ability to harness it effectively is critical for success. Simio's platform leverages this data revolution through data generated and driven models governed by Data Tables. However, the true value of these models is unlocked only through the proper use of Data Tables, making them essential to building robust, informative simulations. As with any modeling process, debugging and verification are critical parts of the development processes. This presentation explores expert techniques for debugging and verifying Data Tables within Simio, demonstrating how tools such as Trace, Watch, Breakpoints, and Notify can be leveraged to realize the full potential of data-driven, data-generated models. By cracking these expert strategies, the model-building process is accelerated to enhance the quality of simulations significantly.

1 INTRODUCTION

Maximizing the potential of Data Tables in Simio is crucial for building structured, flexible, extensible, and reusable models. In these data generated and driven models, the Data Tables control large portions of model logic and object properties. An individual data entry in the Data Table can be dynamically referenced using standard functions. But Row References map specific table rows to Simio Objects and Tokens. When these Objects and Tokens reference a column, the row is inferred from the Row Reference. Simio's Data Tables can follow a relational data model using Key Columns and Foreign Keys. When related tables are connected, setting a Row Reference for one table also automatically sets the reference for the related rows in the connected tables. In a Data generated and driven model, verifying that the right data is being referenced is crucial. Additionally, complex modeling logic involving multiple components and their Row References can make it difficult to know and understand which reference will be used. This work focuses on how to use four tools to debug and verify Related Tables and Row References.

2 TOOLS

Many different tools are available to Simio users to help debug and verify Row References. The first step in using any of these tools is understanding that the Row Reference Resolution can only come from one object and that object is chosen based on a hierarchy. Row references will be checked starting at the top of the hierarchy, and if the object currently being checked does not have a row reference, it will be skipped. There are also three different types of Row References: Table Reference, Explicit Selections, and Potential Rows. Table Reference refers to the absolute row number in the table and can be implicitly assigned. Explicit Selections will show the row that the logic in the model has deliberately set. Potential Rows occur when the object has a row reference to a Key and that Key has many Foreign Keys in other tables.

One of the most useful resources in Simio is the Trace option. Trace shows the detailed steps as the model runs, allowing users to see which objects are referenced. A setting in additional options turns on Table Row Resolution, which shows Row Resolution in a step when a table reference is resolved.

Depending on the amount of data in the model, using the Table Row Resolution Trace can inflate the model, making it run slower and making it more difficult to search through the steps. Debugging best practice is inserting a Breakpoint or a Notify add-on step before the suspected problem area. Using this practice, one would not turn on the Trace or not turn on the Table Row Reference option in the Trace until the Breakpoint is hit or until the Notification alert from the Notify step. Notify and Breakpoints can also help eliminate pieces of logic that are not part of the problem.

Breakpoints and Notify steps can be used similarly; however, there are a few key differences. Breakpoints can be put at specific places in the model that will pause the model when an Entity or Token reaches the breakpoint. Notifying is an add-on process step with more functionality than pausing the model. It can be used in Trace with a specified message without stopping the model. It can be used as a warning, which, when encountered, opens a pop-up window containing a specified message. Once this pop-up opens, users are given the option to pause or stop the simulation. The pause options allow users to turn on the Trace information or access Watch, a tool discussed in the next paragraph. Finally, the Notify step also has the option to act as an Error, which stops the simulation when it is encountered and displays an error with a specified message.

Finally, the Watch window allows the user to view the values of an object's states, functions, and elements during the run. The Watch window also includes Processes and their active tokens under their parent object and child object information within a hierarchical parent object. The Watch Window is only available when the simulation run is paused. Watch can be helpful for quickly checking the Row References of an Entity or Object in the Facility without having to sift through Trace. It is important to note that knowing which object is relevant to the Row Reference is important to use watch to check the reference. Otherwise, it can be used as a good indicator that a different object may be used to resolve the logic.

3 CONCLUSION

This work details the features in Trace, Watch, Breakpoint, and Notify that can be used to understand Row Reference Resolution better. Knowing more about these tools is not only a crucial skill for Related Tables but will also improve debugging and verification skills in other areas of Simio. Equipped with knowledge and tools, challenging model contexts become more manageable. Building Data driven and generated models will become a new standard. Simio believes that Data-driven modeling is such a powerful paradigm that it is the standard practice for all models developed at Simio. Also, leveraging data-driven techniques and relational data lays the foundation for Data-generated modeling using Custom Objects and Simio Templates. Harnessing the power of data is easier than ever through Simio’s features and functions.