AUTOMATIC CREATION OF SIMULATION MODELS FROM DATA MODELS OF LOGISTIC AND MANUFACTURING PROCESSES

Shabnam Tauböck[1], Felix Breitenecker[1]

[1]Vienna University of Technology
Department Simulation
Wiedner Hauptstrasse 8-10
Vienna 1040, AUSTRIA

shaby@osiris.tuwien.ac.at

Abstract

More and more simulation projects are no longer designed as single solutions but for longer and repeated use. The aim is not to simulate one specific scenario but to compare different approaches, even varying system parameters. The idea is based on the arising need for simulation in logistics as well as supply chain management. The most difficult aspect therein lies in the gap between need and knowledge. The users are mostly not familiar with the concept of modeling and simulation and have no prior knowledge of using a simulation tool. Their main interest is not the simulation itself but the results calculated.

To minimize the effort of redesigning the simulation model the approach of self constructing simulation models proves itself to be the most convenient. It enables even users with no knowledge of the simulation tool to change the simulation model by altering the according parameters. Such self constructing models need longer to develop but are a lot more convenient in usage.

The simulation environment is 'hidden' behind a GUI, enabling the user to enter all data specific for the simulation model as well as the simulation run in a certain, user friendly way. The same GUI can be used to start a simulation run and to analyze the results.

This approach implies the split between a simulation environment and a data model; The user only needs to interact with the GUI that stores the data in the structure defined by the data model. The simulation environment imports the data at the start of the simulation run and automatically creates the model. This enables fast and easy comparison of different scenarios while minimizing the effort of parameterization.

Keywords: Enterprise Dynamics, Automatic Model Buildup, Data Model.

Presenting Author's biography

Shabnam Michèle Tauböck works for Profactor Research and Solutions since March 2007. From Dezember 1999 until March 2007 she worked for the business area Process Optimization, Division Environment and Life Science at ARC Seibersdorf research. From June 1998 until April 2000 she worked for PSE SIEMENS in Vienna.

1 General

Simulation is an often used tool for planning and optimization especially for logistic and manufacturing processes. The possibility to compare different solutions to an occurring problem or to observe and analyze the effect of a decision without actually risking a lot of money and time is one of the biggest advantages of modeling and simulation.

More and more simulation projects are no longer designed as single solutions but for longer and repeated use. The aim is not to simulate one specific system with its fixed parameters but to compare and analyze different approaches, even varying system parameters.

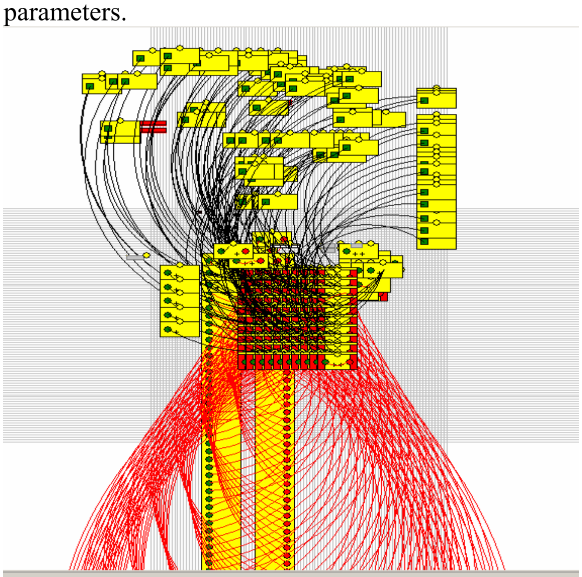

Fig. 1: Connection between Model Elements

There are two aspects that make the use of simulation on an everyday basis more difficult than it might appear:

1. The gap between need and knowledge: Expert knowledge is required to develop an accurate and valid simulation model and it is required to modify it as well. But more often than not this is not existent in the average company as they are not familiar with the concept of modeling and simulation and have no prior knowledge of using a simulation tool. A simulation model is usually developed by external specialists; the main interest for the company is not the simulation itself but the results calculated.
2. The time needed to redesign a model: small changes to a simulation model may not seem too complicated to incorporate; but the more complex a model is, the more time consuming even small variations of parameters may become. If changes to the structure are necessary it can quickly become a difficult task to modify the model. (See figure 1), especially if several different approaches to such changes are to be compared.

During the last years the need for a simulation system to be quickly and easily modified has shown in several different projects for different reasons. The comparison and analysis of different approaches and systems has been one of them, the reusability of a simulation model another. One project included the simulation of 140 different Points-of-Sale; it would have made no sense to create 140 models, instead one generic model was used to simulate each of them as a parameter variation of the basic model.

2 Enhancing the Model Flexibility

A first step to offering a higher flexibility is to allow the changing of parameters within certain boundaries to observe the effect on the whole system.

The next step would be offering a possibility to change parts of the simulation without actually needing to go down to the programming level.

And finally the last development would be a simulation that is based upon certain structural parameters that define the whole system. Using these parameters the complete simulation buildup process can be automated according to the current parameters.

To minimize the effort of redesigning the simulation model the approach of self constructing simulation models proves itself to be the most convenient. It enables even users with no knowledge of the simulation tool to change the simulation model by altering the according parameters. Such self constructing models need longer to develop but are a lot more convenient in usage.

The basic idea is to separate the basic model from the scenario specific data. Keeping the data outside of the simulation environment e. g. in a database offers several advantages:

- the data is much easier to manage
- the data is much easier to modify
- the simulation model is less burdened with storing the data during the simulation run; this is especially noticeable with result data

The data needs to be structured to represent a model of the simulated system that contains all possible combinations of parameters.

The interaction with the database can be easily done via a GUI, hiding the simulation environment as well. Additionally this GUI can be used to start a simulation run and to analyze the results.

This approach implies the split between a simulation environment and a data model: The simulation environment contains the basic model elements as well as the functions to create the simulation model according to the data in the external database.

ISBN 978-3-901608-32-2

The data model contains all additional information needed.

3 Data Model

The data model contains the information needed to create all possible scenarios. This data model is implemented in an external data source, as a database or Microsoft Excel. This data source is then used to specify the scenario to be simulated.

3.1 System Elements

The Data Model contains all elements that may exist in the simulated system. If they exist in a certain scenario depends on the according system structure parameters.

3.2 System Structure Parameters

These parameters define the structure of the simulation model in a certain scenario: of which elements it will exist, how many of them and how they are linked.

3.3 System Parameters

These Parameters define the state of the system.

3.4 Input Parameters

These parameters are relevant for the simulation run; they contain all information that is not directly linked to a certain element.

e.g.: the length of the simulation run

4 Simulation Environment

4.1 The Template

Before the simulation model is created several actions have to be taken. These are executed by functions embedded in the simulation model template.

Such actions are the import of the data from the data model; the creation of the simulation model, the start of a simulation run, the processing of the result data. All these actions can be automatically triggered if a simulation run is started.

4.2 The Simulation Library

The simulation Library contains all elements that are defined in the data model. Each has its own behavior that can be influenced by the parameters given in the data model. The functions creating the simulation model will use these elements and set their individual parameters after the model buildup.

4.3 The Simulation Model

The Simulation Model is the end result of executing the functions for the model buildup and represents the structure defined in the chosen scenario that is based on the data model. All elements have their predefined behavior in dependency from their parameters as specified in the data model.

The System State is defined by the System Parameters; the simulation run is executed according to the input data received from the data model.

5 Example

The aim of this specific simulation project was to develop a simulation to check if certain criteria concerning about 140 different Points-of-Sale were fulfilled. The main focus concerning the results was the duration of a purchase of a certain accessory: the duration starting from exiting the vehicle until re-entering and installing the purchased product must not exceed ten minutes in more than 98 percent.

For the data model MS Access was used, the simulation was done with Enterprise Dynamics. The data exchange was based on the ODBC Interface.

5.1 Basic Model Elements

A Point-of-Sale may be a roadhouse, a gas station or a shop selling this specific product. All components that might occur are defined in the data model and used to form the different scenarios. For example Petrol Pumps are not mandatory: a single scenario could contain between 0 and n petrol pumps.

The simulation library contains these elements with their predefined behavior.

The layout shown in Figure 1 for example, shows a Point-of-Sale consisting of five parking spaces for trucks, one cash and two vending machines.

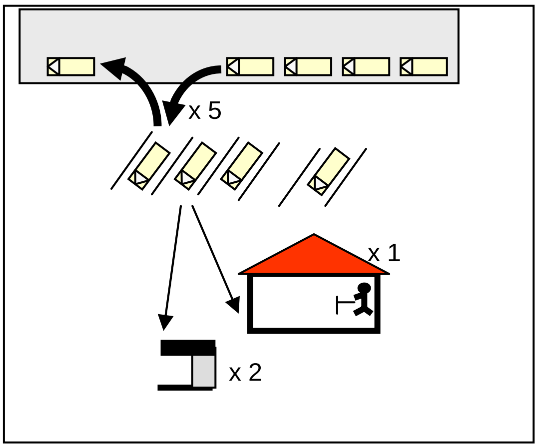

Fig. 2: Layout of a Point-of-Sale

5.1.1 Parking Spaces

Parking spaces for trucks and other vehicles are strictly separated.

Parameters:

- Number of parking spaces for trucks and for cars
- Distances from selling point

- Capacity of queue in front

5.1.2 Petrol Pumps

Similar to the parking spaces there are different petrol pumps: for trucks, for cars or a combination of both. The number of these pumps may be set individually for each Point-of-Sale.

Parameters:

- Number of petrol pumps of each kind
- Distance from selling point
- Capacity of queue in front

The time needed for fuelling depends on the vehicle and the kind of pump used:

- Car at petrol pump for cars
- Truck at petrol pump for trucks
- Car at petrol pump for both
- Truck at petrol pump for both

5.1.3 Selling Points

Basically there are two different kinds of selling points: manned and unmanned. More precisely the product may be sold either in shops or vending machines. Some Points-of-Sale may offer both possibilities. Additionally the manned selling points may have three different kinds of cashes:

- Cashes specifically set up to handle all transactions regarding this product. No other transactions can be done there.
- Regular cashes – no transactions regarding the product can be handled there.
- Combined cashes: all kinds of transactions can be handled there.

This entails that customers that wish not only to buy the specific product have to either use the combined cash or line up twice. The decision which cash to use depends on the length of the queue in front, where the combined cash - if needed - is favoured up to double length.

Additional parameters to be considered:

- Regular traffic
- Opening hours
- Staff resources

5.2 Input Parameters

Flow of Traffic: Due to the rather complex flow of traffic it is divided in several input streams:

- Regular Traffic
- Traffic generated by customers buying the specific product.
- Traffic generated by customers needing any kind of support for the specific product.

It has also to be taken into consideration, that the flow of traffic is also dependent on the time of day. At night there will be surely less traffic than during the day, especially during lunchtime a high number of customers is expected at roadhouses. To account for this the day is divided in different time zones, whereas for each of them a different input distribution may be defined. The number of input streams as well as the number of time zones is arbitrary.

Access Road: If the dimensioning of the parking spaces is not sufficient a queue will form here. It is assumed that overtaking is not possible.
Handling Times: The handling times at cashes and vending machines depend on the kind of customer to be served.

Opening Hours of cashes

Down Times of vending machines

Timeframe: By default a time frame of 24 hours is simulated, it may be extended to any length. .

6 Database controlled execution of scenarios

To ensure a high flexibility the simulation implemented in Enterprise Dynamics is completely controlled by the data model stored in the ACCESS database. Depending on this data the model structure is formed, the single components are connected and parameterized.

The database contains all needed specifications for each Point-of-Sale:

- the number of input streams as well as their parameters,
- handling times,
- the number of parking spaces,
- petrol pumps,
- cashes,
- vending machines,
- the length of queues,
- opening hours

The outcome is a highly specialized simulator, that is able to generate any Point-of-Sale consisting of the specified components, it only has to be defined in the database. Using it to simulate places where one could not be sure if they would be able to deal with the additional customers allowed a better assessment of the future situation, as well as an easy way to evaluate the effect of changes.

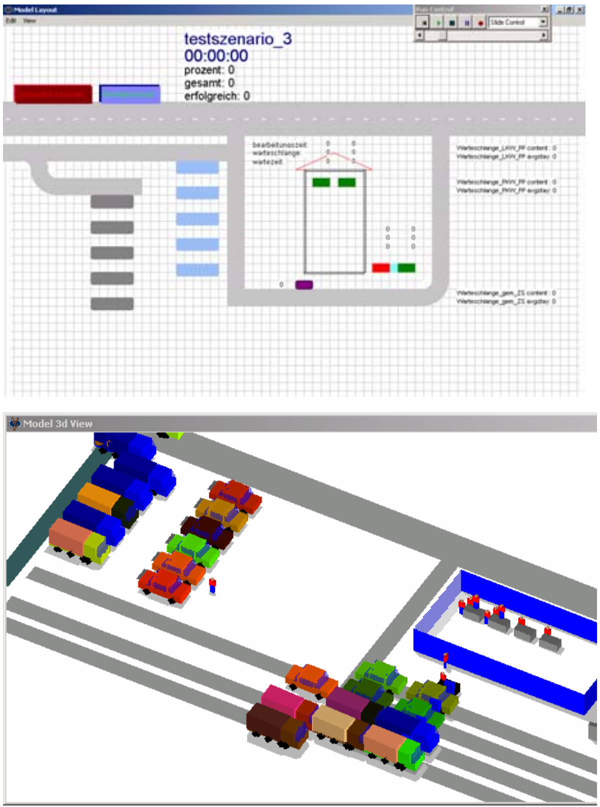

Fig. 3: The Model in Enterprise Dynamics

7 Database controlled simulation

To ensure a high flexibility the simulation implemented in Enterprise Dynamics is completely controlled by the data stored in an ACCESS database. Depending on this data the model structure is formed, the single components are connected and parameterized.

The database contains all needed specifications. A list containing all of the scenarios stored in the database enables the user to choose which ones to simulate. A function implemented in Enterprise Dynamics allows exporting the result data back to the database before starting the next simulation run.

The outcome is no longer a simulation, but a highly specialised simulator, that is able to generate any scenario consisting of the specified components, it only has to be defined in the database. Using it to simulate places where one could not be sure if they would be able to deal with the additional customers allowed a better assessment of the future situation, as well as an easy way to evaluate the effect of changes.

7.1 Library

Each element already contains the full functionality and behavior, every model element created inherits this functionality. For example each kind of cash has its parameters like cycle time, opening times; they are individual for each object. This data needs to be defined.

7.2 Basic System Parameters

As in the prior example parameters for each component are stored in a database, allowing easy and quick editing. With this data the behavior of each single component is well defined.

The model has to be well defined in order to be created accurately. So not only each component of the model is defined via its parameters but also the simulation itself. In our example one of these parameters would be the number of number of parking spaces.

7.3 Functions

To generate the final simulation objects are built from the elements in the library, their individual behavior being influenced by their parameters.

The basic simulation only consists of functions that:

- import the data from an external source
- create the needed components from the library
- parameterize them
- connect them
- reset and initialize the system

The data has to be imported into the simulation, objects have to be created and parameterized according to this.

Now we have a simulation, automatically created basing on the data provided. Changing this data causes a change to the model; therefore the simulation is newly generated. This offers a great flexibility. Once the simulation is implemented scenarios can be tested with hardly any limits to the range of parameters.

Of course, creating the basic simulation, programming the functions that in turn will create the final simulation is more intricate for everything has to be done completely generic to ensure the adaptability to any scenario.

8 Result Aggregation

Most Simulation Tools as Enterprise Dynamics do offer some basic routines for analyzing result data. But as soon as results are used to compare different scenarios or model approaches data needs to be stored outside of the simulation tool.

The results generated in the simulation run are automatically stored for further processing. It is possible to add the functionality to automatically create and simulate a new scenario if certain results are not within given boundaries. The parameters of the new scenario are set according to predefined rules – i.e. if the average waiting time in a queue is to long an additional production line is added to the system.

9 CONCLUSION

The object oriented approach to simulating discrete process flow models offers a new flexibility especially in creating and testing different scenarios. Using the concept of classes and objects enables one to quick changes in the basic structure of a simulation. The effort can be minimized in programming functions to automate the generation of a model.

This makes working with a simulation much more comfortable. Instead of creating several simulations for different scenarios it is sufficient to develop the whole system once, and only parameterize it anew. This furthermore enables even people without much knowledge in developing and working with the software to fully exploit the advantages of simulation.

10 References

[1] Bernhard P. Zeigler, Herbert Praehofer, Tag Gon Kim (2000). *Theory of Modeling and Simulation - Integrating Discrete Event and Continious Complex Dynamic Systems*, Academic Press.

[2] Bauer Heinz. *Wahrscheinlichkeitstheorie*.Walter de Gruyter, Berlin; New York.

[3] Barry L. Nelson.(1995). *Stochastic Modeling – Analysis & Simulation*. McGraw-Hill Inc.

ISBN 978-3-901608-32-2