

CONTINUOUS IMPROVEMENT APPLIED TO SIMULATION MODELING : A CASE STUDY

Arun Jayaraman
Jason A. Green
Ali K. Gunal

Production Modeling Corporation
3 Parklane Boulevard, Suite 910 West
Dearborn, MI 48126

ABSTRACT

A typical transfer machine consists of several synchronous stations served by a common transfer mechanism. Many simulation languages and simulators, available today, may provide useful constructs that allow detailed modeling of such mechanisms. However, large models involving many entities can be slow to run. This paper presents several techniques that can be used to build accurate and efficient models of systems that include one or more transfer machines and long conveyors. The techniques described in the paper were developed during a simulation study of a large-scale manufacturing system. The overall approach to the modeling also demonstrates the application of the "Continuous Improvement" concepts in simulation model development.

1 INTRODUCTION

Many manufacturing facilities operating in a mass production environment make use of transfer machines for high volume, repetitive operations. From a modeling perspective, a transfer machine is not very much different from a single machine that can handle multiple parts. However, in a high volume production environment, model size can grow considerably if several transfer machines each comprising several stations are involved. The execution speed is usually proportional to the number of entities in a model. Consequently, building an accurate and efficient model of a high volume production system with transfer machines becomes a challenge regardless of the modeling language.

The objective of this paper is to discuss some of the concepts involved in modeling transfer machines using WITNESS. A case study involving the simulation

analysis of a manufacturing facility that makes parts for engines will be discussed. The system consists of several machining, testing, and assembly operations, part transfer mechanisms, and buffers between stations. The problems encountered in analyzing the system and approaches taken to tackle them are discussed in detail. Special emphasis is given to the solution methodology as it also demonstrates the application of the Continuous Improvement philosophy to simulation modeling. Although WITNESS is the modeling language used in this study, the concepts and the methods discussed in the paper are applicable to most modeling situations regardless of the simulation tool.

Continuous Improvement is a philosophy widely practiced in manufacturing and quality circles (Suzaki 1987). Also known as, "Kaizen" it is the philosophy of achieving major enhancements in a process through incremental improvements. As the name implies, it relies on the idea that there is no end to making a process better. Each incremental improvement constitutes one of the infinitely many phases of development. Originally used for enhancing manufacturing processes, the philosophy has gained considerable popularity recently, and has been extended to all aspects of business including the software industry.

Simulation modeling and analysis seems to be a promising area for the application of Continuous Improvement principles. For successful application of simulation, a combination of good usage of software, a strong understanding of the process, planning, creative modeling, experimental design, and intelligent analysis must be utilized (Ulgen, Black, Johnsonbaugh, and Klungle, 1994a and 1994b). Clearly, by performing upfront analysis and planning, many of the problems that can arise in later stages of a simulation study can be successfully avoided. However, it is not always possible,

in the first attempt, to build an efficient model that is also an accurate representation of the real system. Thus, most simulation models lend themselves to improvement for efficiency and/or accuracy.

The remainder of this paper is organized as follows. In the next Section, the simulation study that involved the modeling of transfer machines is introduced. The real system and objectives of the simulation study are discussed. Also, an overview of the general features of transfer machines is given. In Section 3, the development of the WITNESS model in several stages is explained. Modifications to the system through simulation analysis and the benefits from the study are briefly discussed in Section 4. Finally, conclusions of the paper are given in Section 5.

2 THE SYSTEM AND THE SIMULATION STUDY

The system consists of two identical machining lines. Each line consists of the following series of operations: 1. rough grind, 2. rough bore, 3. bolt hole, 4. primary wash, 5. bushing press, 6. final grind, 7. final bore, 8. drill, 9. final gage and hone, 10. wash. Parts are transferred between operations on conveyors. In addition to moving parts, conveyors also act as buffers. Operations 2, 3, 5, 7, 8, and 9 are transfer machines. Thus, there is a total of twelve transfer machines.

A transfer machine can typically be viewed as an operation consisting of several stations. A transfer bar is used to move parts from one station to the next, using a reciprocating motion. In a typical machine cycle, the transfer bar moves parts only when the operations of all stations are completed, resulting in a synchronous operation mode. Since a perfect balance of operation times at all stations is usually difficult to reach, residual idle time at some stations may remain. It is possible to avoid idling of the stations by incorporating separate transfer mechanisms between each pair of stations. However, the cost of such a design may outweigh the cost of idling some stations.

Each transfer machine in the system under consideration consists of five or more stations. Each station can process two or more parts simultaneously. Similarly, in each cycle, the transfer mechanism transfers a batch of parts between stations. The synchronous mode of operation also requires that a new batch of parts be loaded at the first station at the beginning of each cycle (to avoid running an idle cycle). Furthermore, when a station in the transfer machine breaks down, all other stations are taken down immediately after their current operation is completed. Transfers between stations are forbidden until the broken station is repaired.

The primary objective of the simulation analysis was to design the system for meeting the forecast demand for the product. The primary performance measures were cycle times of operations, transfer times, and buffer sizes. The independent parameters included preventive maintenance schedules and unscheduled breakdowns across all operations. The preventive maintenance schedules and unscheduled downtimes were based on organizational standards and historical data, respectively.

3 MODEL DEVELOPMENT

As the system under consideration showed a fair amount of complexity, the corresponding simulation model included many elements to capture some of the details of the system operation. However, the focus of this Section is on various ways of modeling transfer machines in WITNESS. The following discussion assumes that the reader is familiar with some of the basic concepts of simulation modeling in WITNESS. More specifically, the model described here makes extensive use of the following WITNESS modeling elements: PARTS, MACHINES, CONVEYORS, BUFFERS, and VARIABLES. For a detailed description of these elements and many others not mentioned in this paper the software documentation provides an excellent source (AT&T ISTEEL Limited 1994).

As mentioned earlier, as with any other software product development, a simulation model can be developed at different levels of detail. As more detail is included in the model, the execution speed decreases. The model of the system described in the previous Section was developed in several stages to obtain a balance between model accuracy and execution performance. The initial version of the model included the most detailed representation of the system. However, as many experiments would be needed to analyze the system, the execution speed quickly became an issue as a single run of the model took more than eighteen hours. After establishing a baseline analysis of the system with the detailed model, the study focused on improving the model performance. The following is a description of the various stages of model refinement to obtain faster execution speed.

3.1 Stage 1

In this initial attempt to model the system, each station of every transfer machine is modeled explicitly by using a MACHINE element. For the sake of discussion, the MACHINE element that corresponds to station i on the transfer machine will be called ST(i). The transfer bar is also modeled by using MACHINE elements. Each piece

of the transfer bar between two consecutive stations is represented by a MACHINE element. For discussion purposes, the MACHINE element that represents the piece of transfer bar between stations i and j will be called TB(ij). Then, it is clear that there would be a total of eleven MACHINE elements to completely model a transfer machine with five stations: one element for each of five stations and six elements for the transfer bar. These six elements for the transfer bar would include four machine elements modeling the transfer bar between the five stations, one machine element modeling the transfer bar loading the first station and one machine element modeling the transfer bar unloading the last station.

To represent the movement of parts in batches, two or more parts are batched into a single part at the first station. As the batch moves along the transfer machine, it is delayed at each station for the cycle time of operation. The transfer times between machines are also represented as delays at machine elements representing pieces of the transfer bar. Batches finishing the operation on the last station are then split into individual parts before moving onto the next process.

The synchronous nature of the transfer machine is captured in the model by coordinating the instances when batches can move. A batch, called b_1 , moves into ST(1) and is delayed for the cycle time of the station. Then, it moves into TB(1,2) representing the transfer bar between the first and second stations. As soon as another batch, b_2 , moves into the first station, b_1 moves from TB(1,2) to ST(2). The two events described above happen synchronously. After b_1 completes its operation at ST(2), it moves into TB(2,3) representing the piece of the transfer bar between second and third stations. In a similar fashion, b_2 transfers into TB(1,2). Then, b_1 and b_2 wait for the next batch to be loaded into the first station. As before, at the same simulation instant, b_1 and b_2 are transferred into ST(3) and ST(2), respectively. The same logic is extended to all other stations and transfers between them.

When a station goes down in the model, all batches wait in elements TB(1,2) through TB(4,5) representing the transfer bar. The batches start moving only when all the stations become operational again. This fact forms the basis of the improvements explained later.

Between operations, the parts are moved on conveyors. Besides moving parts between operations, these conveyors also act as buffer storage areas. For example, when one operation goes down, the succeeding operations can continue working by pulling parts out of the buffer. Similarly, the machines before a broken machine can keep pushing out parts onto the succeeding conveyor until it is full. In this modeling stage, these materials handling/buffers are modeled using

CONVEYOR elements. An advantage of this modeling method is that all parts in transit between operations can be seen to move smoothly on the conveyor. While this feature provides a significant benefit from the animation standpoint, a large number of CONVEYOR elements in a model can be detrimental to the execution speed of the simulation.

3.2 Stage 2

There are several characteristics of transfer machines that can be used for simplifying the model. First of all, there is only one input point to the transfer machine and one output point from it. Also, all stations start their cycles at the same time. Even if a station completes its cycle faster, it cannot transfer the processed part and receive a new part unless all stations in the transfer machine are ready to cycle. Furthermore, the transfer machine does not release a batch of parts unless a new batch is available to the first station. Another important feature is that the downtimes at a station will force all the stations in the transfer machine to come down.

These observations lead to modeling a transfer machine as a single MACHINE element with one input and one output point. The cycle time of this machine would be equal to the longest operation time of any station of the transfer machine. The cycle time would also include the time to transfer a batch from one station to another. By using past data it would also be possible to determine a downtime distribution for the entire transfer machine. Scheduled downtimes (such as tool changes and flushing the system) were deterministic occurrences applicable to the transfer machine rather than individual stations.

An immediate benefit of this modeling scheme is the reduction in the number of MACHINE elements in the model from 132 to 12. An important drawback of this modeling method is that the residence time in the system is underestimated. However, since the objective of the simulation study was to determine the throughput of the system, the approximations described above did not have a significant impact on the results. Furthermore, comparisons of the results with those obtained from the first (the more detailed) model showed that the improved model estimated the throughput accurately.

3.3 Stage 3

Having simplified the modeling of the transfer machines without sacrificing functionality, the modelers then devoted attention to reduction of other material handling elements in the model. As mentioned earlier, the large number of high speed conveyors reduced the execution speed of the model. Furthermore, the simulation

analysis made to this point in the study showed a need for larger buffers between operations. This increase in buffer size meant longer conveyors between operations, and hence, it resulted in an even slower run-time performance. These long conveyors were not only detrimental to the execution speed, but they also appeared cumbersome in the animation. To remedy this situation, the external buffers were modeled as combinations of CONVEYOR elements and BUFFER elements. Thus, there were short segments of conveyors going into and out of each operation. Buffer elements were introduced between these short elements. Each part coming out of an operation was first transferred onto a conveyor segment. Each part, then, traveled over the length of the short conveyor segment and moved into the buffer. Once inside the buffer, each part was delayed for the time required to travel over the length of conveyor being modeled. Those travel times could be easily calculated by using the conveyor length and speed. This conversion of the CONVEYOR elements into BUFFER elements improved the run-time performance of the model without degrading the quality of the animation.

3.4 Stage 4

Once a reliable model of the system was obtained at the end of stage 3, further efforts were made to improve the execution speed of the model by converting all conveyor elements into buffer elements. The travel time on conveyors were calculated as explained above. Then, these travel times were used as the time delays in BUFFER elements that represented the conveyors. This change further improved the model performance. After all the modifications have been made as explained in Sections 3.2 through 3.4, the execution speed was nearly double that of the first and more detailed model.

3.5 Stage 5

The execution speed of a WITNESS model is also proportional to the number of temporary entities that exist in the system at any time (parts constituted the temporary entities of this model.) This is particularly true when there is a large number of entities in the model that stay on conveyors (or in buffers, in general) for extended periods of time. Consequently, any improvement that will reduce the number of temporary entities in a simulation model can significantly improve the run-time performance. Keeping this in mind, an alternative way of handling temporary entities was proposed. The proposed approach relied on using variables and arrays to reduce the number of temporary entities. At the time this paper was written, this stage

was not implemented in its entirety. However, the authors have used and investigated this concept in other models. Not only does this modeling technique present a means of improving execution speed, but it also presents a more elegant, transportable, and expandable method for modeling similar, larger systems.

In this approach, a global variable array is defined, representing a specific conveyor between two machines. Furthermore, each array element will hold the instant of time a specific part is transferred onto the conveyor. The size of the array should be no larger than the capacity of the conveyor. In other words, one position in the array should represent one position on the conveyor. By doing this, the model logic determines if a part has been on the conveyor long enough to complete its travel to the next machine. Since an array element needs much less computer space than an entity and much less computer time to process, size and execution speed of the model improve markedly.

The following notation is used in the algorithm described below.

Conveyor_{M to N} refers to the actual conveyor joining machines M and N.

Array_{M to N} refers to data concerning parts on Conveyor_{M to N}.

Count_{M to N} is a counter representing the current number of parts on Conveyor_{M to N}.

Max_{M to N} refers to the maximum number of parts that can be on Conveyor_{M to N}.

Travel_{M to N} refers to the travel time on Conveyor_{M to N}.

Batch_M refers to batch size of parts on Machine M.

Batch_N refers to batch size of parts on Machine N.

X_M is a counter corresponding to specific part in a batch of parts at Machine M.

X_N is a counter corresponding to specific part in a batch of parts at Machine N.

Machine M is capable of pushing a part to Conveyor_{M to N} only if the following condition is true:

- Conveyor_{M to N} is not full and is capable of handling batch of parts at Machine M ($\text{Count}_{M to N} + \text{Batch}_M \leq \text{Max}_{M to N}$).

If the above condition is true, the following steps will be executed in the model:

1. Set $X_M = 1$;
2. Wait until Machine M completes processing next part;
3. Record the time (current time) in $\text{Array}_{M \text{ to } N}$ when the X_M^{th} part is being pushed onto $\text{Conveyor}_{M \text{ to } N}$;
4. Increment $\text{Count}_{M \text{ to } N}$ by one;
5. Increment (modulo the array size) the array subscript to be used for the next time-stamp;
6. Scrap the actual part entity corresponding to X_M that is in Machine M;
7. If $X_M < \text{Batch}_M$
 $X_M = X_M + 1$
 Goto Step 2
 Else
 Goto Step 7
 Endif
8. Pull next batch of parts into Machine M and go back to Step 1, provided that
 $\text{Count}_{M \text{ to } N} < \text{Max}_{M \text{ to } N}$.

Note that if the entities have attributes that they must retain, i.e. color, size, etc., a multi-dimensional array could be used for $\text{Array}_{M \text{ to } N}$.

Operation N is capable of pulling a part from $\text{Conveyor}_{M \text{ to } N}$ only if the following condition is true:

- $\text{Current Time} - \text{Array}_{M \text{ to } N}(\text{Current Index})$
 $\leq \text{Travel}_{M \text{ to } N}$

If the above condition is true, the following steps will be executed in the model:

1. Set $X_N = 1$;
2. Pull a new entity into the model from world to represent a part being removed from $\text{Conveyor}_{M \text{ to } N}$; (Note: In WITNESS, "world"

refers to any area external to the system under consideration. All defined parts not being used in the model, currently, exist outside the system, in the world. These parts are available to the system when needed and can be "pulled" in.)

3. Decrement $\text{Count}_{M \text{ to } N}$ by one;
4. Set time-stamp at $\text{Array}_{M \text{ to } N}$ (Current Index) = 0.0;
5. Increment (modulo the array size) the array subscript to be used for checking the next time stamp;
6. If $X_N < \text{Batch}_N$
 $X_N = X_N + 1$
 Goto Step 2
 Else
 Goto Step 7
 Endif
7. Begin processing parts at Machine N.

Note that if a multi-dimensional array was used for $\text{Array}_{M \text{ to } N}$, the corresponding values of the entity attributes can also be retrieved from it and can be assigned to entities.

Clearly, this approach can dramatically reduce the total number of entities in a model that includes long conveyors that can hold many parts. Consequently, substantial improvements in the model execution speed can be expected. (the authors plan to report comparative results in the final version of this paper)

4 RESULTS OF THE SIMULATION STUDY

The simulation analysis helped predict the system throughput for optimal combinations of operation times, material handling system speeds, buffer sizes, and preventive maintenance and breakdown schedules. The results also showed that the forecast demand for the product could be met by replacing the two machining lines with only one machining line, after appropriate adjustments in the design. These adjustments included increasing the capacity of some of the conveyors and reducing some of the machine cycle times. Considerable cost savings were achieved by using one of the lines

instead of both. However, increasing the capacity of the conveyors led to long conveyors between operations, which might occupy valuable floor space. This problem was solved by using multiple lane conveyors as opposed to single lane conveyors.

5 CONCLUSIONS

The concept of "Continuous Improvement" presents a philosophy that is being applied increasingly to all aspects of business. Simulation analysis, with its multifarious activities, requires careful planning of time and resources and lends itself well to continuous improvement in all phases of its life-cycle. A careful application of the philosophy can improve all aspects of a simulation study. However, generating model code can benefit the most from continuous improvements as can any other software development effort. As demonstrated in the paper, one way of utilizing the philosophy is by developing the model in distinct stages. By doing so, a reliable model exists throughout the entire study starting from earlier stages. This approach may seem to contradict other approaches that require extensive upfront planning and design to avoid expensive changes in model design. However, it should be noted that it is almost mandatory that a simulation study should start with a good model from the beginning for any improvements thereafter to be successful. Consequently, a careful balance between i) upfront planning and design before modeling starts and ii) making improvements after an initial model is completed should be sought when building simulation models.

ACKNOWLEDGMENTS

The authors thank Dr. Onur M. Ülgen, President, Production Modeling Corporation, MI and Professor, Department of Industrial and Systems Engineering, University of Michigan, Dearborn for his encouragement and comments. The authors would also like to thank Mr. Edward J. Williams of the Ford Motor Company, Dearborn, MI for his comments.

REFERENCES

- Onur M. Ülgen, J. J. Black, B. Johnsonbaugh, R. Klungle. 1994a. Simulation methodology in practice - Part I: Planning for the study, *International Journal of Industrial Engineering*, Volume 1, Number 2, pp. 119-128.
- Onur M. Ülgen, J. J. Black, B. Johnsonbaugh, R. Klungle. 1994b. Simulation methodology in practice - Part II: Selling the results, *International Journal of Industrial Engineering*, Volume 1, Number 2, pp. 129-137.
- WITNESS User Manual. 1994. AT&T ISTEEL Visual Interactive Systems Ltd, United Kingdom.
- Donald G. Askin and C. R. Stanridge. 1993. *Modeling and analysis of manufacturing systems*, John Wiley and Sons, Inc., New York.
- Kiyoshi Suzuki, The new manufacturing challenge. 1987. - *Techniques for continuous improvement*, The Free Press, New York.

AUTHOR BIOGRAPHIES

ARUN JAYARAMAN is an Applications Engineer at Production Modeling Corporation, Dearborn, MI. He received a B.S. degree in Mechanical and Production Engineering from Annamalai University, India and he received an M.E. degree in Industrial and Systems Engineering from Virginia Polytechnic Institute and State University in 1990 and 1993, respectively. His consulting interests include Discrete Event Simulation analysis and training, MTM analysis, KANBAN based systems, automotive manufacturing and domestic appliance manufacturing systems.

JASON A. GREEN is a Systems Consultant at Production Modeling Corporation, Dearborn, MI. He received his B.S. degree in Industrial and Systems Engineering from the University of Michigan - Dearborn in 1992 and his M. S. E. in Industrial and Systems Engineering from the University of Michigan - Dearborn in 1994. His consulting interests include Discrete Event Simulation analysis and Finite Capacity Scheduling systems in the manufacturing field.

ALI K. GUNAL is a Systems Consultant at Production Modeling Corporation. He received his Ph. D. degree in Industrial Engineering from Texas Tech University in 1991. Prior to joining PMC, he worked as an Operations Research Specialist for the State of Washington where he developed a simulation system for modeling and analysis of civil law suit litigations. At PMC, he is involved in consulting services for the analysis of manufacturing systems using simulation and other Industrial Engineering tools. He is familiar with several simulation systems including AutoMod, Arena, Quest, Robcad, and Igrip. He is a member of INFORMS, IIE, and SME.