ANALYSIS OF THE PRODUCTIVITY IN STRUCTURAL STEEL FABRICATION PROCESSES

Evaluation and comparison of different Simulation-Software for the analysis and optimization of production processes at steel fabricators

A. Hofacker[1], D. Gandhi[2]

ABSTRACT

Structural Steel Fabrication is a process in which there is a wide variety of product mix and where most of the products are unique in nature. A lot of work has been done to analyze the use of Discrete Event Simulation in various manufacturing and construction projects for application of lean concepts and other purposes to detect opportunities for optimization. However, there is a lot of different Discrete Event Simulation (DES) Software available in the market. After a first evaluation, 5 different software tools were selected that all claim to identify bottlenecks in processes and optimize or improve them. This paper specifically takes the complexity of Structural Steel Fabrication Process into consideration and uses these 5 different Software Simulation tools available to model these processes. It not only shows the need for using simulation for better production planning and layout-decisions during execution to improve productivity, but also compares these Softwares, in order to help Simulation practitioners to make informed decisions on software selection based on the mentioned parameters for comparison.

KEY WORDS

Discrete Event Simulation, Steel Fabrication, Structural Steel, Process modeling, Software Comparison, Productivity analysis, performance metrics

INTRODUCTION

Steel is being used in most of the construction projects because of the flexibility, speed of erection and the economic benefits that it provides for the designers and the contractors (AISC 1998). Structural Steel is mostly fabricated in controlled shop environments off-site, for better precision and accurate detailing and manufacturing. The different parts are then grouped together as modules at steel-fabricators and then assembled on the construction site.

The complexity of Steel Fabrication process is originated in the uniqueness of steel projects like bridges, car parks, power-plants, etc. This makes processing, routing and resource or labor requirements different for each product within a steel fabrication shop floor. Hence, Steel Fabrication is a project based industry with low repetitiveness in production and a diverse range of products.

Although knowing all the complexities involved in structural steel fabrication, currently still in most fabrication shops planning, scheduling and estimating activities are left totally on the personal experiences of Production engineers and managers. The only tools used by most fabrication shops for estimating and scheduling are CPM/PERT along with the available

[1] Research Associate, Institute of Technology and Management in Construction (TMB), Karlsruhe University (TH), Germany, Am Fasanengarten Geb. 50.31, D-76128 Karlsruhe. Tel: (49) 721 608 5476, Fax: (49)721 695245, Hofacker@tmb.uni-karlsruhe.de

[2] Bachelor Student, Department of Civil Engineering, Indian Institute of Technology Guwahati, North Guwahati, Assam – 781039, India. Tel: (91) 9957817470, g.chetna@iitg.ac.in

Product drawings and the knowledge about the current status of the shop floor (Abourizk, Song and Wang 2005). However, these tools offer a very limited opportunity to evaluate the different possible scenarios that take into account the many options that each product has across the shop floor and also the way in which available resources are utilized. Moreover, these tools don't show waste and non-value adding time. Hence they are not adequate to manage the production flow, to detect potential for production improvement and to optimize the performance of the material flow.

Simulation tools can imitate the operations of real-life systems or processes to almost any amount of detail required. The ability of these tools to model various products, interdependencies between operations, use of resources, routing and also taking into account various uncertainties makes them suitable for modeling structural steel fabrication or any manufacturing process. Computer Simulation also allows quick modification of major project parameters for the purpose of analyzing different options for optimization without need for real-life experimentation. Currently there are many Simulation softwares that are easily available and which help in achieving process improvement, identifying bottlenecks in existing processes among other things mentioned above. The objective of this paper is to evaluate these different softwares on a certain set of parameters so that potential users and practitioners can decide with ease the software best suited for them after weighing the different evaluating parameters. Also we reinstate the fact that for the need for continuous improvement in a highly competitive environment, steel fabricators should further adopt sophisticated tools of analysis like computer simulation.

RESEARCH FRAMEWORK AND OBJECTIVE

An extensive online-literature analysis has been conducted to identify the relevant software programs for the purpose of analyzing the productivity and bottlenecks in the steel fabrication processes. The objective of the study is embedded in a bigger research framework of elaborating different methods to evaluate and optimize the productivity of production and logistics processes, related to industry sectors at the edge between manufacturing and construction. Besides organizational analysis which is based on qualitative measures, the focus of this paper is to provide with simulation tools, further possibilities to quantitatively model and optimize production processes. Therefore, approximately 50 different simulation softwares were taken into consideration in a first step. Out of these, 5 softwares were selected for comparison among the many others available. Hence these Simulation softwares were selected due to their ease in availability, recommendations from different experienced simulation users and the softwares' past expertise in simulation.
The 5 softwares that are dealt with for comparison are namely: Flexsim, Siemens Plant Simulation, Arena, Simio and Extendsim. Input parameters and calibration data is obtained by several production site visits done by the authors. After having modeled the general production process, the authors went back to a steel fabricator to obtain further detailed data to feed the model, to discuss the results and obtain feedback from practitioners.

THE GENERAL PRODUCTION PROCESS OF STEEL-FABRICATORS

Structural Steel Fabrication is a process where a steel piece, say a Heavy Plate or a Beam is cleaned and then detailed (i.e. cut, drilled, etc) as defined by the fabrication drawings. The Fabrication drawing provides all the details about the dimensions of a component to be fabricated, and it´s location to be fitted together with other components. After the detailing process, different components are fit together manually with pointed welding-fits, again as given in the drawings. Then the product is semi-automatically welded for acquiring the required strength. Afterwards it is cleaned for preparing a smooth surface, painted for surface protection and then shipped to the construction site for erection. Figure 1 shows the typical main production process in Steel Fabrication.

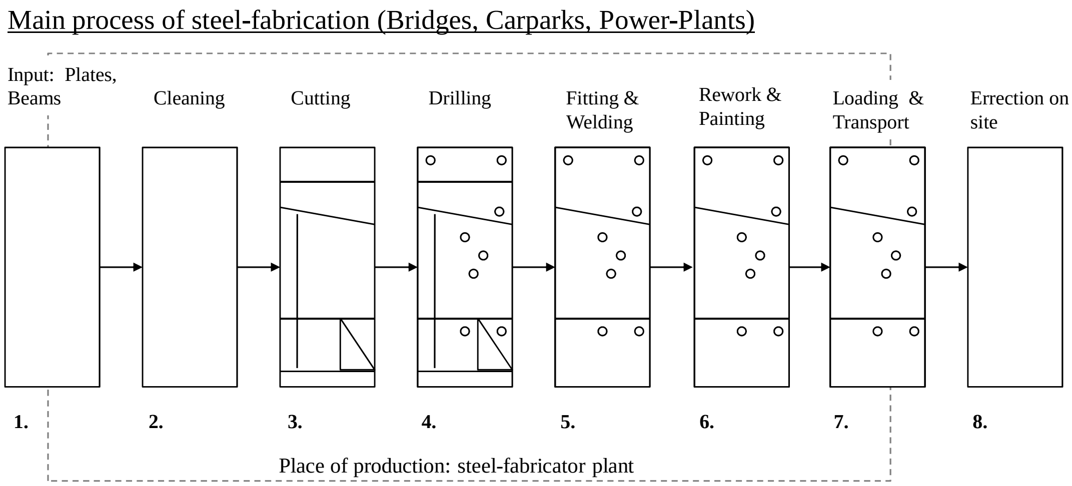

Figure 1: Main production process of steel-fabrication

There are typically areas between each activity (work in process inventory). Also there is a certain amount of security stock of input material kept (beams, heavy plates, tubes). Inspection is done after welding and painting to ensure quality.
The process described above is very general in nature and there can be exceptions depending upon the requirements of a specific project.

DISCRETE EVENT SIMULATION AND MODELING THE PROCESSES OF STEEL FABRICATORS

The difficulty in modeling steel fabrication processes is found in the nature of these production processes that comprise elements of discrete events as well as continuous flow. Steel fabricators measure productivity in terms of tons produced and not as number of components produced, due to the low degree of repetitiveness of each product component. Thus it is important to create a link between the number of components and tons when using discrete event simulation.
In this section, the general processes as mentioned above are modeled in a Discrete Event Simulation Software for the purpose of comparison of the different software tools. The following description assumes that the reader is acquainted with the basic terminologies involved in using simulation software.
In the simulation study two different type of entities are taken as input and processed through the system, namely, 'HP' (Heavy Plate) and 'B' (Beam). For this there are two different objects as entry points named 'HP' & 'B' which send the entities in the model according to the mentioned dispatching schedule. As these entities arrive in the model, they are stored in a queue object (Queue1) to simulate the incoming of trucks with the ordered items. These incoming entities are transferred from the 'Queue1' by an object named 'Crane1' used to simulate the delay that occurs in unloading the items from the truck by a Crane into the Storage area. The Storage area is modeled using a queue object named 'Storage_Area'. Again using processing delay objects named 'Crane2' & 'Crane3' for the time taken to transport the entities from the 'Storage_Area' to the first processing station named 'Cleaning'. The object

‘Cleaning’ represents a Cleaning Machine in the shop floor where entities go through getting cleaned and have a time for getting processed.
After getting cleaned the entities are put in a temporary storage area named ‘Queue2’ and then are routed to their respective detailing machines. The entities ‘HP’ & ‘B’ are routed to the detailing machines named ‘Cut_HP’ and ‘Cut_B’ respectively. Here the ‘Cut_HP’ & ‘Cut_B’ simulate CNC machines which carry out the required detailing operations on the entity. The uncertainty related to the processing time and the number of parts in which each entity is cut into is dealt by incorporating statistical distributions. There is always a certain amount of wastage involved with detailing of the entities, hence we route the wastage from ‘Cut_HP’ & ‘Cut_B’ to queues named ‘Waste_HP’ & ‘Waste_B’ respectively while the remaining cut parts are placed in the queue named ‘Queue3’.
The cut parts are then collected by the processing stations named ‘Fitter1’ & ‘Fitter2’ to fit the different cut parts as mentioned in the fabrication drawings together. The uncertainties related to fitting variable number of parts and the time taken for fitting different parts together is again dealt by specifying statistical distributions for both. Finally after the parts are fitted together, they are placed in the temporary storage area named ‘Fitted_Items’. The model ends with the operation of fitting and does not include the next down-stream process steps of Surface Preparation, Surface Protection and Shipping. This is because of simplification and overview reasons, but also because these three process-steps are similar to the ones already modeled before. The objects to be used for model building in each of the software are different and have different names; however, the underlying logic to be applied is very similar.
The model described above is a very simplified and a very basic model which is used in a first step only for the purpose of comparison of the 5 DES software and showing the possibilities to simulate the required processes also to a greater level of detail. In reality there are many Cranes, Cleaning, Cutting and fitting stations and other resources in a Steel Fabrication shop-floor. For example the complexities related to a CNC machine’s processing time for each piece can be simulated by creating a library of different processing machines which incorporate different processing times and machining options based on the fabrication drawings and detailing of different pieces. In this model, any resources such as laborers and operators are not used for simplifying the model in order to compare the different simulation tools on the same platform with the same input data. Again, there are many uncertainties relating to the laborers required for fitting each product and the time for fitting also varies on factors like labor productivity, size & geometry of the product to be fitted, etc. All these complexities can be simulated to quite an extent by using statistical distributions and detailed knowledge about the influencing factors involved, provided there is sufficiently precise measured input data.
Thus within these conditions of simplification, it is seen that that Discrete Event Simulation is a powerful tool to simulate the current conditions of the fabrication shop to a great level of detail.

Input Data Table for Comparison:

The same input data was used for the processes described in the previous section for the purpose of comparison of the different softwares. Table 1 describes the data used in the different objects of the model. Run time for all models is taken as 4000 Minutes, which represents approximately one week of operation, depending on the number of working hours and operation shifts per day.

Table 1: Input data for model calibration and comparison of simulation programs

OBJECT NAME	TIME per event (minutes/event)	CAPACITY	INPUT	OUTPUT
HP (heavy plate item)	10	1	1	1
B (beam item)	8	1	1	1
Queue1 (unloading truck)	-	100	From HP + B	FIFO
Crane1 (transportation time of material to stock)	Uniform (2 , 5)	1	1	1
Storage_Area (security stock)	-	10000	From Crane1	Random
Crane2	Uniform (2 , 7)	1	1	1
Crane3	Uniform (2 , 7)	1	1	1
Cleaning	Uniform (1 , 3)	1	1	1
Queue2	-	100	From Cleaning	FIFO
Cut_HP	Uniform (8 , 10)	1	1	Uniform (10,30)*
Cut_B	Uniform (8 , 10)	1	1	1
Waste_HP	-	1000	10% of Cut_HP	To Sink
Waste_B	-	1000	5% of Cut_B	To Sink
Queue3	-	10000	90% of Cut_HP + 95% of Cut_B	Random
Fitters 1 & 2	Uniform (10 , 30)	30	Uniform (10 , 30)	1
Fitted_Items	-	1000	From Fitters 1 & 2	To Sink/Exit

PARAMETERS USED TO COMPARE DISCRETE EVENT SIMULATION (DES) TOOLS:

The 5 parameters built for the basis of comparison to test the adequacy of the DES Softwares are:

1. Visualization.
2. Detail/Accuracy in Simulation.
3. Level of Coding Required.
4. Analyzing tools.
5. Documentation of Reports.

Rating System for DES Software comparison:

A rating system for comparing the Softwares was defined for proper evaluation of the parameters. The DES Softwares will be rated on a scale of 1 to 6, where a rating of 6 on a certain parameter will be given to the Software which satisfies the parameter definitions to the highest level compared to the other four softwares on this parameter. Thus a rating of 1 implies a poor standard for the parameter and a rating of 6 implies the software being very good in that particular evaluated parameter. The requirements of each user may be different and one may develop his own rating system based on the description of the parameters for individual software.

1. **Visualization:** Here visualization implies the animation which is available to view the model, so that the built model resembles real life objects and conditions. This is an important parameter to evaluate different DES Softwares as animation can be of great benefit in enticing people in the organization to be interested in process improvement using Simulation when they can actually see how the model is working as they see in real life. [rating example: 1=…, 6=…]

2. **Detail/Accuracy in Simulation:** Different DES Softwares incorporate different levels of detail to simulate a model. They also differ in their capabilities to model the uncertainties involved in real life systems. They have different sets of in-built objects in their libraries with which a system can be modeled. Thus it is important to know which software can simulate the process to the required level of detail with their available in-built tools, so that model-making is quick and easy.[rating example: 1=...; 6=...]

3. **Level of Coding Required:** Each DES Software generally has its own programming language. This coding language is provided in order to give the modeler the flexibility to create his own objects and other features as required using coding. This feature is particularly of great use to experienced modellers who have been using that specific software for many years. For new users it is usually impediment and time consuming. In the parameter "Level-of-Coding" it is analyzed the type of required input data and whether the simulation tool has sufficient inbuilt features or whether it requires further coding for building the basic model; (if coding is necessary, then what is the level of knowledge of the programming language required). [rating example: 1=...; 6=...]

4. **Analyzing tools:** Once the processes have been modeled in a particular DES Software and once the simulation is run, how can one be sure about the results that are shown? So here, by using the term 'Analyzing tools' it is evaluated whether the Software provides enough support to carry out the 'What-if' Analysis and check how a changes made in a particular process affect the rest of the model. Often Softwares provide analyzing tools such as an optimizer to optimize a certain project parameter such as total profit. They also have the option of putting in costs for operating a machine or to use a resource, so that we can compute the overall profit using the optimizer. Almost all softwares have the ability to show replications of different possible scenarios, if the input data contains statistical distributions. Hence this parameter evaluates the analyzing power of the different Softwares. [rating example: 1=..., 6= ...]

5. **Viewing of Results/Reports:** This important parameter for comparison describes the ability to see and plot all the results at one place and in different forms (bar charts, pie charts, graphs...) once you are finished building, running and analyzing the model. This helps to better understand what actually happened and eases the explanation to someone else about how the existing facility is working

and accordingly what changes are needed to improve it. [rating example: 1= ...; 6= ...]

EVALUATION OF THE DES SIMULATION TOOLS

In this section, evaluation and rating of the 5 DES simulation tools has been carried out, on the basis of the defined evaluation parameters.

I. Simulation tool: Flexsim[3]

Figure 2 shows the basic model, built and fed with data from table 1. In the following the simulation tool is evaluated in detail on the before mentioned 5 parameters.

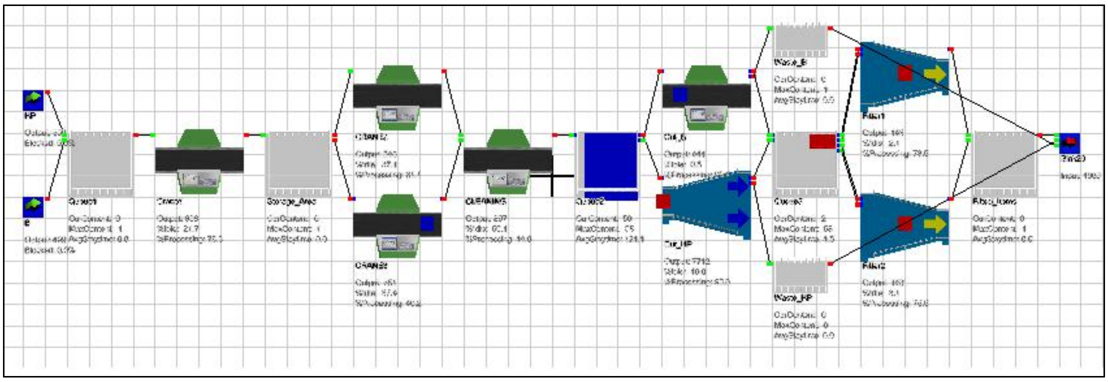

Figure 2: Basic Model in Flexsim

1. Visualization: This simulation tool provides a 3D modelling environment with its objects resembling real-life. It has in-built animated objects that are dynamic and act as a superb visual aid in understanding how the final system will perform. Hence, there is no extra time needed for the visualizing the model's animation. As mentioned before a good animated model is very effective in getting management's attention and influencing their way of thinking. (Obtained rating score: 6)

2. Detail/Accuracy in Simulation: The objects available in its standard library are sufficient to simulate our basic model and enable also to simulate higher level of details for more complicated models. There are also many options to simulate logistics flow such as conveyors, cranes, transporters and operators, where the network path and distance along with the speed of a transporter can be specified. However, in the linking the objects with attributes in the basic model, coding had to be used for simulating security stock and to specify a statistical distribution for number of items fitted by the object 'Fitter' and the number of items output by the object 'Cut_HP'. (Obtained rating score: 4)

[3] Flexsim trial version 4.52 was used for the purpose of this paper.

3. Level of Coding Required: Flexsim has its own programming language called Flexscript, but also allows modellers to do the coding in C++. As mentioned above the modeler needs to have programming skills for accomplishing their required objective. Hence this is a big drawback for those modellers who don't have any programming knowledge. (Obtained rating score: 3)

4. Analyzing tools: Various analyzing tools such as the optimizer and the financial module are available to input costs for operation and optimize a certain variable. Also an experimenter tool is available which helps to run the model through different scenarios by changing the specified variables and collecting data for each scenario. By using these tools, 'what-if' analysis can be carried out efficiently. (Obtained rating score: 5)

5. Viewing of Results/Reports: Reporting and documentation of results is very comprehensive in Flexsim. The results can be viewed by exporting the required project parameters to an Excel file, or a detailed history of the model on Microsoft Access. Furthermore, various other views such as pie, bar or Gantt charts and graphs are also possible to view with all the required details. Hence this simulation tool provides a good way of presenting the results of the analysis that helps to identify bottlenecks easily. (Obtained rating score: 5)

II. Simulation tool: Siemens Plant Simulation[4]:

Figure 3 shows the basic model as built in simulation tool called "Siemens Plant Simulation (SPS)".

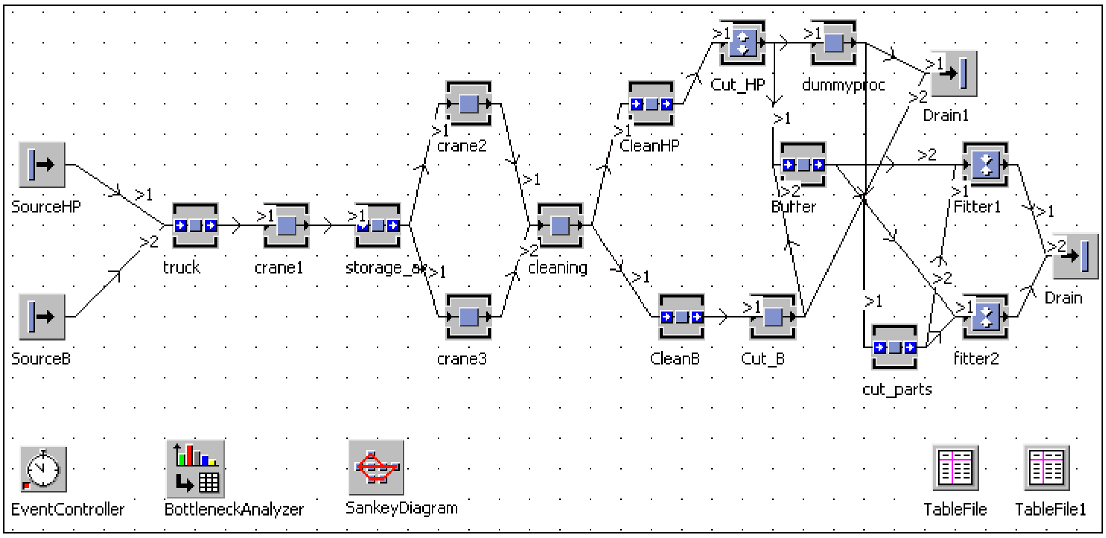

Figure 3: Basic Model in SPS

1. Visualization: SPS provides a good visualization tools to make the model. It has a 2D as well as 3D working environment. This helps to better understand how the model is

[4] Simul8 2009 evaluation version was used for the purpose of this paper.

working and facilitates to explain the finished model to someone else.[here the explanation is still a bit poor and general. Please precise in two more sentences;] (Obtained rating score: 5)

2. Detail/Accuracy in Simulation: The detail in simulation provided by this tool is enough to simulate the basic model with ease. Also more complex models can be built with relative ease and quickness as the number of objects available in its library is very detailed so that they can be used for simulating various processes. No programming or coding was required to build the basic model. (Obtained rating score: 6)

3. Level of Coding Required: SPS uses a ‘Method’ object for programming and making user-defined changes to the way in which standard objects behave. This helps the user to design and alter the behaviour of objects as they desire. As mentioned before, no use of the ‘Method’ object was required in building the basic model. It’s use might be required for more detailed models. (Obtained rating score: 5)

4. Analyzing tools: SPS has standard analyzing tools such as the Optimizer, the financial module and also has the capability to run multiple replications. Apart from these tools, it also has a Kanban module which can be used to simulate a pull system in your model. This helps us to analyze the differences in the push and pull processes and how it affects the overall productivity of the factory. (Obtained rating score: 6)

5. Viewing of Results/Reports: The documenting of the results of one run or of multiple replications is good and easy as compared to the other Softwares. SPS shows the results in terms of charts, graphs and tables. There is no need for the modeller to check the results for each object separately. This is a significant advantage in particular when the model size is large. As mentioned before, the showcasing of results should be well to navigate and comprehend; otherwise it is difficult to be sure about the reliability of the results and decisions. (Obtained rating score: 6)

III. Simulation tool: Arena[5]

The basic model built in Arena is shown in figure 4.

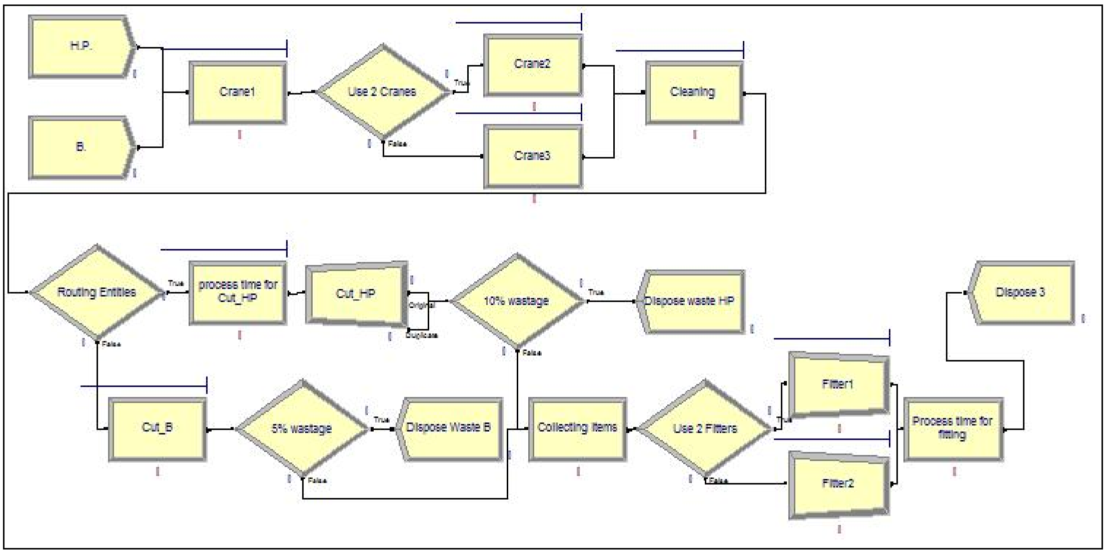

Figure 4: Basic Model in Arena

1. Visualization: Arena uses the flowchart objects and provides a 2D modeling environment. Animation has to be defined and linked to the model separately by the modeller. Thus animating the model after the basic flowchart based model is built consumes considerable time. (Obtained rating score: 3)

2. Detail/Accuracy in Simulation: Arena has an extensive set of libraries containing different objects to create the model we require. A model can be created to any level of detail. However, for simulating the security stock in the basic model it is required to use Arena´s programming language. (Obtained rating score: 5)

3. Level of Coding Required: Arena has its own custom built programming language named 'Siman', which can be used to modify the properties of objects as required. Despite of having a comprehensive library some features like the one of the security stock mentioned above is missing. Hence learning Siman id imperative for building sophisticated models which again requires from new users to learn a new programming language. (Obtained rating score: 3)

4. Analyzing tools: Arena too has an optimizer, a financial analysis module and the option for analyzing various replications and scenarios of a particular model, enabling a proper analysis of the built model. (Obtained rating score: 5)

5. Viewing of Results/Reports: The results can be viewed at the end of each model run and can also be exported to an excel or a pdf file, either in table form or as bar charts. However, the results are not as comprehensive and organized as required to quickly detect

[5] Arena 12 trial version was used for the purpose of this paper.

bottlenecks in material flows. A lot of information is put together at one place and it becomes difficult to determine the relevant information. (Obtained rating score: 3)

IV. Simulation tool: Simio[6]

Figure 5 shows the basic model built in Simio.

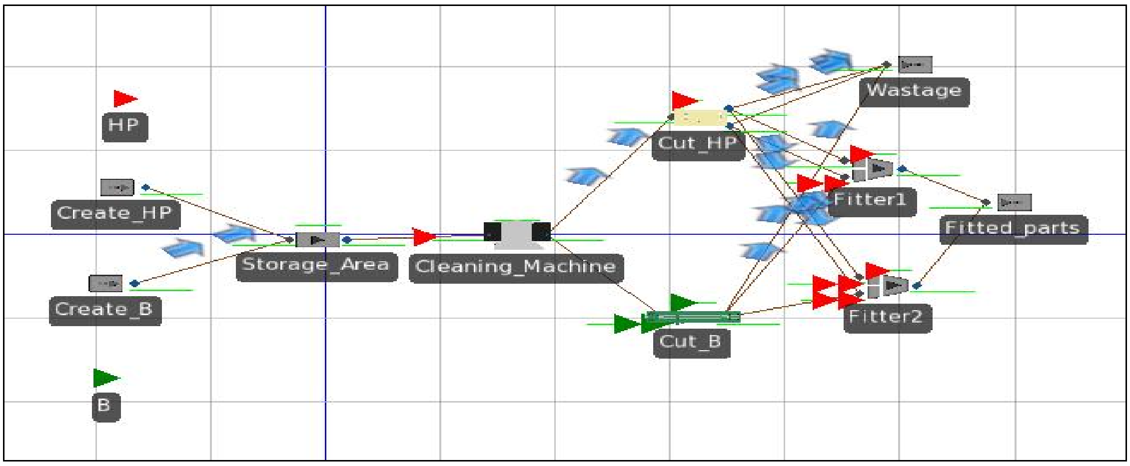

Figure 5: Basic Model in Simio

1. Visualization: Simio offers a 3D modelling environment. The modeler has the flexibility of animating the objects as he wants from their symbols' library or 3D symbols that resemble real-life objects can also be downloaded from the internet and then used for animation. (Obtained rating score: 5)

2. Detail/Accuracy in Simulation: The basic model was built in Simio with the use of its in-built tools only. It offers sufficient details to model the current state of the shop floor simulation of the basic model with input values from table 1. It has a standard object library and provides many other tools to create a model. All these tools and objects provide enough flexibility to simulate the model till the required level of detail. (Obtained rating score: 6)

3. Level of Coding Required: Simio does not have its own coding language which users have to learn but it allows the flexibility of programming in any of the standard programming languages like C#, J#, Visual Basic, etc. Thus a variety of users who know different programming languages can make modifications to the model as they want without the need to invest time in learning a particular programming language. (Obtained rating score: 4) [if no programming is required for the basic model and for other levels of sophistication they provide interoperability with standard programming languages, why did you rate it with only 4 points?]

[6] Simio Release Software 1.0.3182 evaluation version was used for the purpose of this paper.

4. Analyzing tools: Simio's current version does not have any good analyzing tools like the optimizer or taking into account costs for processing, transfer or storage times. Although it has the option to generate multiple replications to view different scenarios but as compared to other tools' data analyzing capacity, Simio offers below average analyzing tools. (Obtained rating score: 1)

5. Viewing of Results/Reports: Documentation of results is not as good as compared to the other 4 simulation tools. There are the standard reports and the option to export to excel, but it does not show results in the form of graphs or charts. Hence viewing the results also consumes a lot of time of the modeller and makes it difficult to understand what exactly did happen within the simulation model. (Obtained rating score: 1)

V. Simulation tool: Extendsim[7]

The model built in Extendsim is shown in Figure 6.

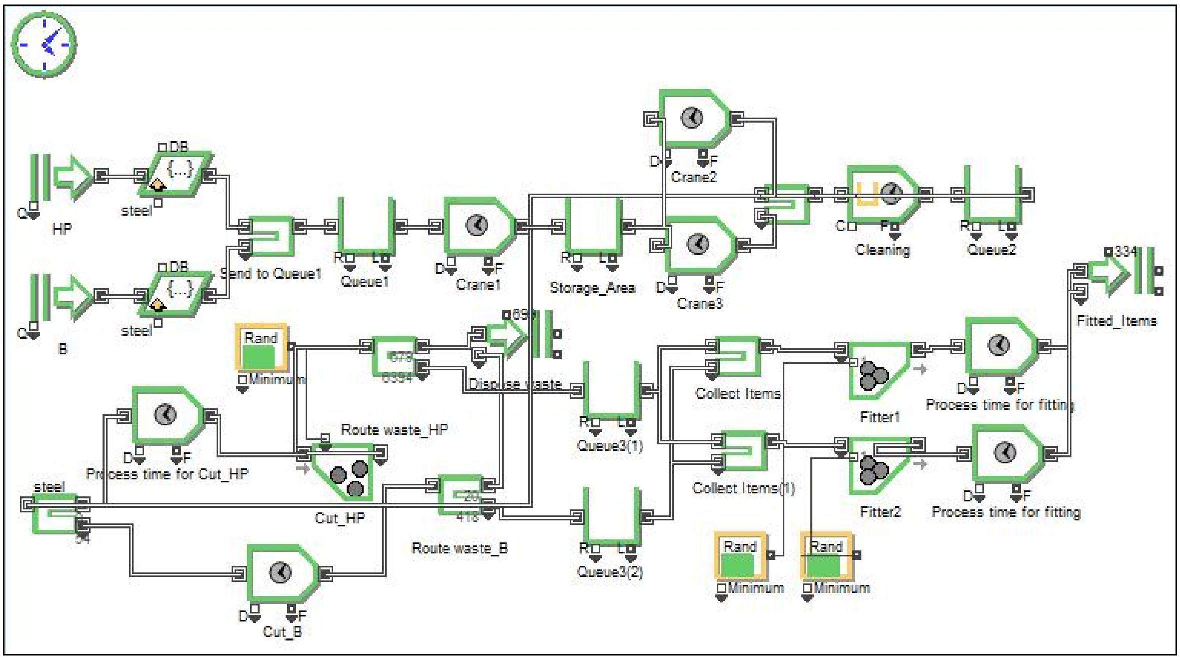

Figure 6: Basic Model in Extendsim

1. Visualization: As shown in the figure 6, Extendsim displays the model in 2D. There is also an option to view the model in 3D, but running the model in 3D reduces the speed of simulation drastically and also it does not resemble real- life objects. (Obtained rating score: 3)

2. Detail/Accuracy in Simulation: Extendsim offers many different libraries and a comprehensive set of objects with which the modeller can create a simulation at a great

[7] Extendsim 7 Demo Player was used for purpose of this paper.

level of detail taking into account various uncertainties. The basic model is built using all its in-built library objects. (Obtained rating score: 6)

3. Level of Coding Required: The simulation has its own programming language 'ModL' to allow the modeller to build his own set of blocks in a user-defined library. As mentioned before the library of Extendsim is already very comprehensive compared to most of the other 4 softwares. Again if the user at some point of time feels the need to create his own set of blocks then learning ModL is imperative. (Obtained rating score: 5)

4. Analyzing tools: Extendsim uses its own optimizer tool, a financial module and also various other custom made analyzing tools like the sensitivity parameter which helps you to determine the effect of changing a certain project parameter on the results of the simulation. Overall it has almost the same set of analyzing tools as mentioned in the other softwares. (Obtained rating score: 5)

5. Viewing of Results and Reports: Compared to the other simulation tools, Extendsim does not deliver the same level of standard in displaying and documenting the results. It shows graphs by using its plotter object or alternatively the results of each object can be viewed by going into the property-values of each object. However it is not possible to visualize overall charts and the tabulation of results is also not well elaborated for the purpose of analyzing bottlenecks in production processes. This makes it difficult to rely on the model built. (Obtained rating score: 2)

SUMMARY AND CONCLUSION

The 5 Discrete event Simulation tools used for this paper are: Flexsim, Siemens Plant Simulation, Arena, Simio and Extendsim. Discrete event simulation is adequate to model the processes of structural steel fabricators with some basic simplification of the processes. An extensive comparative study was conducted on the usability of DES software tools with the purpose of analyzing bottlenecks in production processes of steel fabricators. The five parameters of comparison are namely: Visualization, Detail in Simulation, Coding Required, Analyzing tools and Viewing Results. Thus these DES Softwares were evaluated and rated on a scale of 1 to 6 (a rating of 1 signifying very poor performance- and a rating of 6 signifying the software being excellent in that particular parameter compared to the other investigated Softwares). Figure 7 shows an overview of the ratings given to the simulation tools, considering the five parameters. It should be kept in mind that these ratings are based on the mentioned versions of these Software programs and can change with the newer versions that might come in the future. This paper helps the user identify what type of simulation tool to use for the purpose of simulating processes of structural steel fabrication and what should one look for when making a commitment in using it. However, there is no cost-benefit analysis yet conducted on the acquisition costs of the software versus its adequacy for analyzing production bottlenecks.

In this paper we have used a very simplified and basic layout of a structural steel fabricator and modelled this same layout in all the Simulation Softwares. We also have used the same input values for all the models to verify the consistency of all Softwares. It was observed that each software had some advantages. The use of Simulation technology is new for Steel fabricators in Germany. Simulation requires detailed measurements of existing system and accurate input data to find out bottlenecks in existing systems and also for checking the results on modifying the material flow. Bottlenecks can be found in DES Softwares by checking if stock is growing in the process-steps in front of a particular activity. In real-life systems however, stock cannot grow beyond a certain extent because of the constraint on the available space for storage, but the bottleneck still limits the overall productivity. By carrying out various experiments through simulation it is possible to analyze the productivity, determine bottlenecks quantitatively and simulating options to reduce these bottlenecks.

After this thorough evaluation the authors will continue detailed simulations analysis with SPS on steel fabricators processes. What we feel most important is how the results are shown once the model has been built, as this helps is verifying the accuracy of the model built and identify the bottlenecks easily. This in turn helps in taking correct decisions at the right time, which is finally the aim of using these tools. However, depending on the particular focus of the researcher or manager, there can be preferred different simulation tools, for which the paper provides useful comparisons on different important parameters.

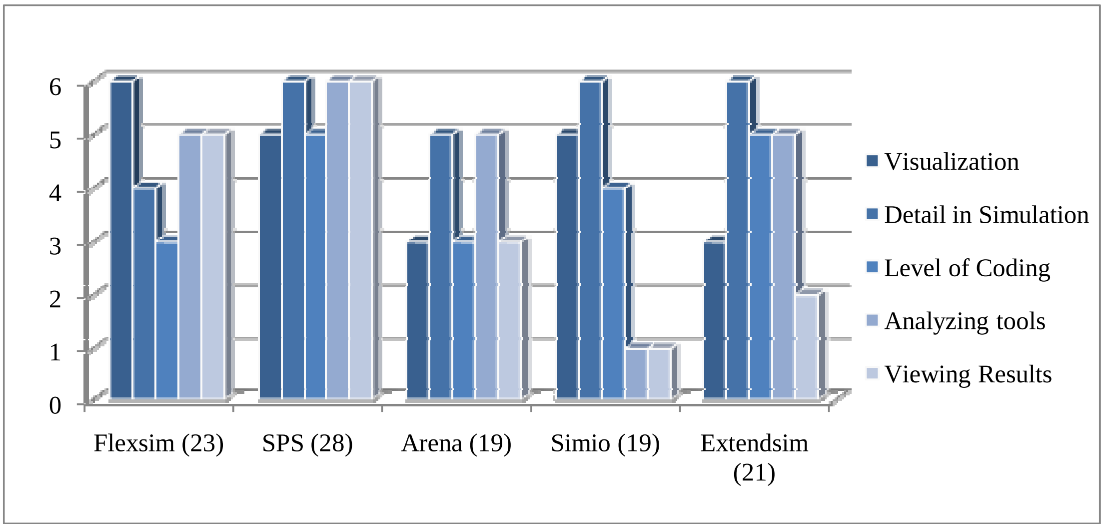

Figure 7: Rating of the 5 DES simulation software programs on the 5 parameters

FUTURE OUTLOOK

In this paper the need for the use of DES Softwares for analyzing and calculating improvement options on productivity of steel fabricators was justified. Future research is proposed to be continued by using adequate simulation tools to create a more detailed model of an actual steel fabricator with precisely measured input data. A case study is proposed in association with a steel fabricator, to simulate its existing factory floor and to further verify and validate the use of DES Softwares for quantitative production analysis. In addition to this, research will be conducted on evaluating lean-management methods and their influences on the productivity of production processes, taking into consideration different production-flow methods like Kanban and pull principles, which are not yet sufficiently implemented or tested in steel-fabrication process in Germany.

REFERENCES

Abourizk, S.M., Song, L., Wang, P., (2005) A Virtual Shop Modeling System for Industrial Fabrication Shops, Simulation Modeling Practice and Theory, Volume 14, Issue 5, July 2006.

Detty R.B., Yingling J.C. (2000), Quantifying the benefits of conversion to lean manufacturing with Discrete Event Simulation: a case study, International Journal of Production Research 1366-588X, Volume 38, Issue 2, 2000.

Goodson, E., (2002) Eine Fabrik unter die Lupe nehmen – schnell und effizient, (Rapid-Assessment-Plant) Harvard Business, Vol 6

Schriber, T.J., Brunner, D.T. (2005), Inside Discrete Event Simulation Software: How it works and why it matters, Proceedings of the 2005 Winter Simulation Conference.

Soriano-Meier, H., Forrester, P.L. (2002) A model for evaluating the degree of leanness of manufacturing firms, Journal of Integrated Manufacturing Systems, Vol 13/2, page 104-109

Tommelein, I.D., Discrete Event Simulation of Lean Construction Processes, IGLC-5 Proceedings.

Womack, J.P. and Jones, D.T. (1996), Lean Thinking, Touchstone Books, London

Womack, J.P., Jones, D.T. and Roos, J. (1990), The machine that changed the world, Macmillan, New York, N.Y.