A SHORT GUIDE TO CLIENT'S SATISFACTION FROM A SIMULATION MODEL

Michal Stec[(a)]

[(a)]SimulateFirst, Stefan Meier Str. 167, 79104 Freiburg, Germany

[(a)]stec@simulatefirst.com

ABSTRACT

After 10 years of applying simulation I come to the following conclusion: the top four features of a simulation model, which guarantee client's satisfaction are: ease of use, good animation, automatic model generation and full results.

I show a decent way, which has a relatively high chance to lead every simulation project to a success. The only success factor that matters is client satisfaction, measured by the number of projects done for returning clients.

Keywords: simulation model, animation, ease of use, automatic model generation

1. INTRODUCTION

All the discrete simulation enthusiasts, me including, know how great the simulation technique is and how much it can potentially offer. However, manufacturing industry has been less than successful in the implementation of simulation as a decision support tool (Heavey 2004). This means practically that it is not easy to find and convince a client to a simulation project.

One could expect that every simulation project is better than the previous one. Thus, with time, experience und support from the simulation organizations and other simulation bodies every simulation user should be well advanced in this field after just a few years. However, simulation technique is still not that widely used in the present world.

My short explanation to that dilemma is that manufacturing clients do not want to try this technique on another project. I mean, they might have used it once but they do not want to use it anymore. Or maybe until another simulation consultant brings some hard facts and convinces them that he can do it in a new, better way. Clients may not even clearly state why they refuse simulation, because the formulated goals of the past simulation projects have been even met. However, they feel that they have not gained enough from simulation. They noticed that the technique is difficult to deal with, they do not trust the results and above all it is expensive.

In this article I am not announcing that I found the golden rule about how to carry out simulation projects. Actually, such rules and guidelines for simulation models have been already described in numerous books, to mention one of the most popular: Law et al (2000) and the summary of how to build credible models in Law (2005). In this article I present a decent practical way, which builds on that simulation knowledge, but which additionally increases a chance of client satisfaction from the simulation project. Client satisfaction is the most important success factor, measured by the number of projects done for returning clients. Only this helps to spread a good word about simulation and guarantees stable development of simulation service.

2. SEEK CLIENT'S SATISFATION

I find that clients are modest in defining goals for a simulation project. After a short presentation they are impressed with examples of my previous models and they think how great the technique is. When we talk about a pilot project they quickly decide that they do not need fancy animation. Instead, they just want a small model with i.e. five machines, which should provide only some simple results. They promise that the model will be extended in the following phases, if only their goals are fulfilled.

Client's assumption is that the simple simulation model leads to a short project timeline, thus with relatively small budget. What he does not know is that the simulation consultant needs to make a relatively big step in learning the client's world, collecting data and drawing processes. The time of such a pilot project is not necessarily short, or if declared short, then there are high chances that it is underestimated, so it lasts longer than planned. After that time the client is no longer satisfied with what he ordered. He expects more.

My top four features that clients always want, although they may not say it up front, are:

- to play with the model (ease of use),
- to see a comprehensive animation,
- to rearrange the model automatically and
- to read the full results down to the smallest detail.

These proposed features make every model a successful one. They simply manifest the power of the simulation tool to the client. The remainder of this paper gives a clue regarding each of these features.

3. EASE OF USE

A simulation model is in fact a standalone computer application, which is designed and developed to bring important findings to the client, fast and meaningful. There are countless applications for making our lives easier and which provide a particular benefit to the user. Simulation could be one of them too. The client would like to use it as he uses Microsoft Office for example. In that case the client would like to use the model even on frequent basis. He expects then that the input data is clearly organized, the animation model doesn't require long time to test one scenario and that the reports are detailed enough to answer his questions. In that case a simulation model could be used even by a simulation novice, but with a good idea what needs to be simulated.

A very good approach to “ease of use” simulation tool is adding a front-end interface to the simulation tool for data collection and report analyses. Such interface could be done for example in Excel from Microsoft. Excel is commonly used by many if not most of decision makers. This fact has been already recognized by many simulation users and simulation software vendors many years ago. So, why not use it as a data keeper for all simulation projects, instead of defining the data in the internal simulation environment. This way the client has one application fewer too learn and he feels like he knows it already for a long time.

	A	B	C	D	E	F	G	H	I
1	Machine	a (m/s2)	a (m/s2)	v (m/s)	v (m/s)	grip	put	from	to
2		a x	a z	v x	v z	s	s	pos id	id
3	Crane 1	2.00	1.00	3.00	1.00	3	3	1	4
4	Crane 2	2.00	1.00	3.00	1.00	3	3	3	6
5	Crane 3	4.00	1.00	4.00	1.00	3	3	6	15
6	Crane 4	2.00	1.00	3.00	1.00	3	3	14	20
7	Crane 5	3.00	1.00	3.00	1.00	3	3	20	25
8	Crane 6	2.00	1.00	3.00	1.00	3	3	25	28
9									
10									

Figure 1: A data table in Excel: machine

A data table in an Excel sheet doesn't have to be fixed to its original size, as it would be in the case of a parameter form in some simulation environment. The tables in Excel can grow wide and long enough for most applications. And all this is done without any work on the development side.

Another advantage of Excel is that it can be used for rapid prototyping of a simulation application. Thus, before sending an offer to a client for a simulation project, the Excel simulation cockpit can be already roughly defined. The client feels then how he will use the simulation model once it is ready. He can press the buttons that will trigger procedure exporting the data to the simulation software. He may see the temporary output data that will be updated after each simulation run. The client sees then what input data is considered and what output will be read from the model. Then he may come up with his own ideas, which can be quickly added to this prototype. This way the two parties, the client and the simulation engineer, come gradually closer to the common understanding of the modeled environment.

How the tables in Excel are organized depends on the model requirements and data hierarchy. I find it useful to draw flowcharts and class diagrams in simplified UML (unified modeling language), but with practice simple data structures can be defined directly in Excel. The more complex the data structures are, the better they should be planned.

Such simulation cockpit becomes a front-end application (1st tier), from which data is transferred to the simulation model (2nd tier). That simulation model may exist then even as a black box to the client, in order to hide its complexity away from him. On the other hand that model should stand out like a neat and well organized structure, in case some quick changes have to be done in front of the client.

Of course, what is easy from the client's point of view does not have to be that straightforward to implement in a simulation model. However, a properly carried out model conceptualization phase (for example in UML) will most likely simplify all client's requirements regarding the project and will help to design a valid simulation model, which receives client's acceptance fairly soon.

The data arrangement in data tables and the relations between the tables are important for the success of a simulation model. All data should be easily accessible from the model level whenever required. This simplifies model construction and enables manual or automatic manipulation with a great deal of parameters. Thus, it makes sense to design the data structure in a way, which resembles a simple database. The rows in tables have their ID, and particular features are referred also with ID. These are, in turn, elaborated in separate tables.

UML is widely used for application development (Fowler 1999). There have been a number of articles written on using UML in simulation (Barijs 2001), a practical tool for model conceptualization phase, providing a dynamic perspective to the information available from modeling tools. It is clear that UML is not “magic” software that will help convince your client to the simulation project. Instead, this tool helps you, as a simulation expert, to plan and design the project structure, so it includes all what the client is asking for. It enables gathering all the data tables, objects of the simulation model and process flows in one place. At this stage it should be possible to answer many design questions, i.e. how are machines, operators and products related to each other? How are they defined in the data tables? Finally how will they be regenerated automatically? This allows to uncover many false assumptions and design pitfalls. This saves time and puts the project on the right track from day one.

The classes from the UML diagram in figure below could be easily represented as tables in the Excel cockpit. This UML diagram has been drawn in Microsoft Visio. Other widely used UML package is Visual Paradigm.

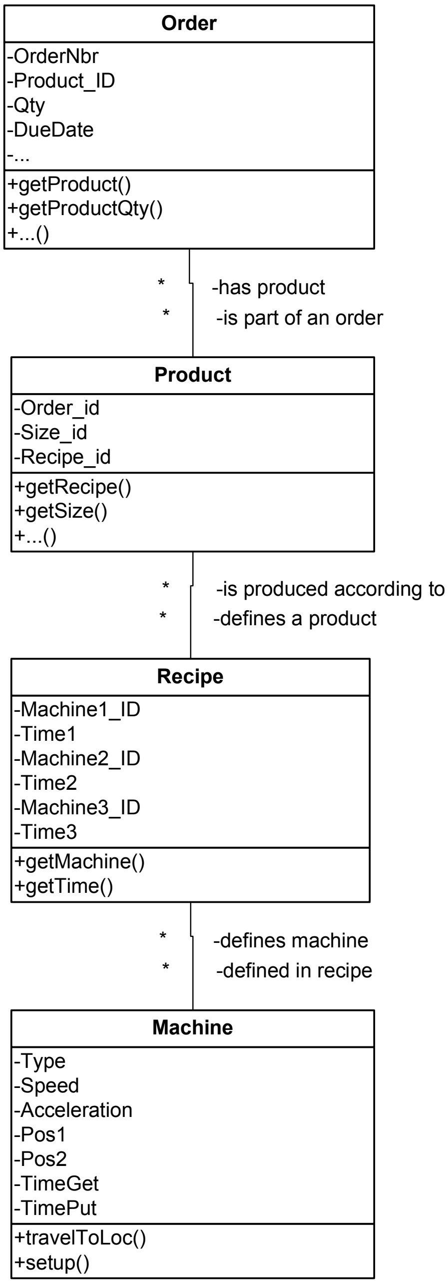

Figure 2: A UML class diagram.

4. ANIMATION

Once a simulation model receives the complete data from the simulation cockpit then it is immediately ready for a simulation run.

It makes sense to help the client understand how the data and logic work together in the model, so he grasps quickly the power of the complex calculations in the simulation model. Explaining the code at this point would make an opposite effect, loose of interest on his side. Instead, the animation does exactly what the spoken explanation would aim for: the client identifies himself with the model; he may run it in a slow pace and check if the model does what required. Without animation the client has little chance to make sure that the model is reliable and does what he expects.

Some models have 2D animation some 3D. Needless to say, the 3D animation is preferred as the client may easier compare the model with what he knows or had seen. The 2D animation is often too abstract, which for the first time viewer is not easy to follow.

The model in figure below is a production machine with a number of cranes moving products to consecutive positions, where the products are processed chemically. The movement of these cranes is not trivial with multiple dependencies of one crane on another. This 3D animation proved here very useful for the client, indeed.

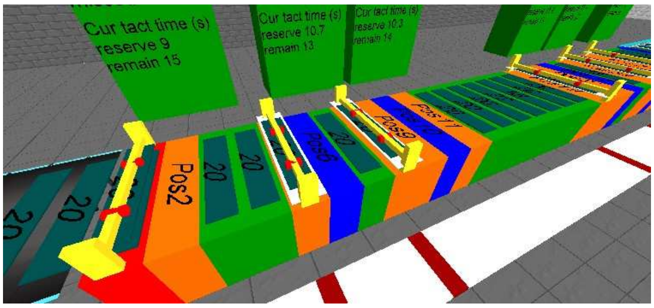

Figure 3: 3D animation of a production machine.

When designing animation for a simulation model it is recommended to ask the client and yourself frequently: what is important in the model, which parameters have to be shown there and how? A good animation usually suffices to get an approval and meet satisfaction level from the client. For example each crane in my example model has a full information table (see figure 4), with throughput, number of serviced parts and other output data.

There are simulation packages on the market, which offer 3D animation built into the simulation environment. Instead of post processing animation type, it becomes rather an integral part of a simulation model. Each time a module is added to the model, it comes with a 3D representation, either drawn from basic 3D shapes or imported from a VRML file. Thus, 3D animation should not take much project time anymore, and it gives a concrete value to the client, which is hard to overestimate.

5. AUTOMATIC MODEL GENERATION

At first the client tries various parameters in experimenting with the model. He changes process times, failure rates and setup times. Soon after that the client asks for adding another machine. And then he hears that this option needs another day to be implemented.

There are obvious extensions to each model, which with little effort come to one's mind: number of resources, areas sizes or paths for traveling objects (Stec 2006). Some of them are simple parameters for existing simulation modules, but other ones are subject to automatic generation of a module in the simulation model.

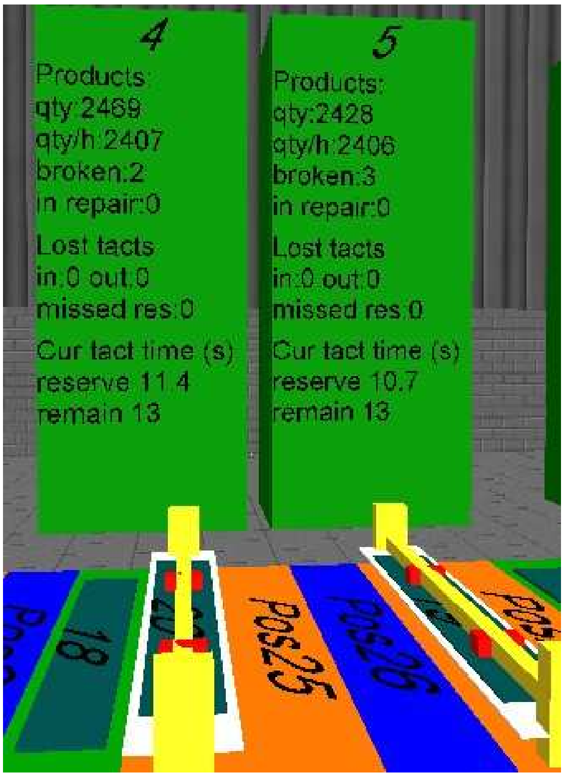

Figure 4: Information table on machine performance in the 3D model.

At first this may look difficult, in some simulation environments this may not be even possible. But when the simulation environment utilizes modules, which can be fully controlled from code, then short programs written in the proprietary simulation script will do miracles. Moreover, with some practice this appears to be even a better modeling technique than the manual “drag and drop” of the modules followed by manual definition of the module parameters. It is because once the scripts are written they can be reused multiple times.

The Excel cockpit described in the section 3 of this paper has numerous data tables defined for user input. Assuming that there is a “machine” table in the cockpit, the client has a possibility to enter the parameters for the predefined machines in the model, whereas each machine is defined in one row of the table. But then the user may simply want to add another machine to the table, because it looks as easy as filling in another row with another machine data. If so, then why not allowing him for adding it to the model on the fly?

A machine is defined as an object or a module. It has a number of parameters such as location, size, speeds, accelerations, processing time, and many others. The machine has also built in operations like setup, processing and similar. All these parameters and operations can be set and triggered from the simulation script level, thus diminishing the need for manual drag and drop of the modules.

The parameters and type of machine operations to perform are defined in the machine table in the Excel cockpit (see figure 1). Thus, all what is needed is a simple VBA interface in Excel to transfer the data to the simulation model, which, if necessary, will also recreate missing machines there.

Automatic model generation improves greatly model maintenance. This enforces that the model structure is well thought over. Adding statistics or another parameter is no longer an issue. It may require changing only one line of code and regenerating the model.

In the example shown in figure 3, the machine was supposed to run with 30 positions and 6 handling cranes. The simulation experiments shown that actually 32 positions and 5 cranes would produce much better result. Adding the new positions, removing one crane and updating a number of connection and parameters would require normally an hour and a few tests. But, thanks to the automatic model generation, the new simulation model was built within a minute. Needless to say, the client was thrilled.

Last but not least simulation is perceived by many clients as an expensive tool. That expense reduces significantly with every consecutive use. Thus, the client will be always more eager to spend his money on simulation project if the model could be reused on another logistic center, another shop floor in another plant or new factory layout.

The ease of use factor plays again a major role. The easier it is to build a virtual plant in the simulation tool, the less effort it takes to convince the decision maker to use the tool again.

6. FULL RESULTS

After the initial validation of the model based on the model animation, a client may want to check what happens in the model once it is simulated over a long period. The client certainly does not want to test it with real time animation then. Thus, the results collections come here in place very handy. In addition to the usual statistics like min, max, avg, stdev and half width, the user may want to see a plot with what actually happened minute after minute in the simulation. This way he gains an additional confidence in the simulation tool. He might for example see what a particular crane in the model was doing. This could be explained for example by a state diagram like this one below.

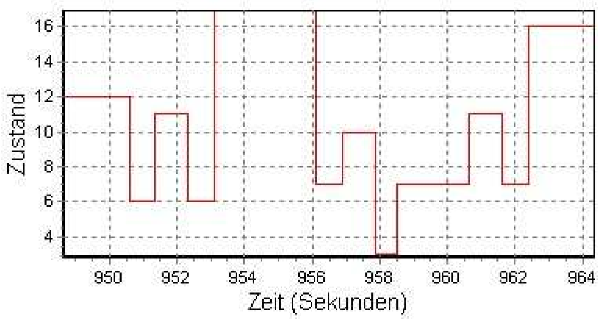

Figure 5: State diagram.

Another useful method for reading detailed results is Gantt chart. The one in figure 6 shows resource occupation by a product over the whole simulation time. One color in the chart below represents one product ID. One can zoom in and out in this chart to the required detail level.

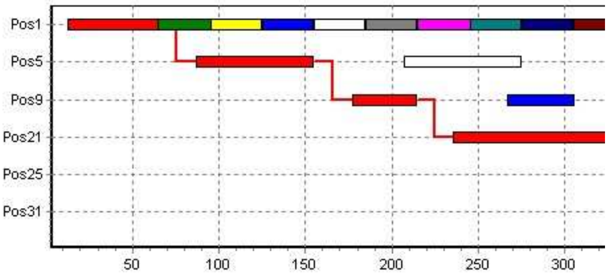

Figure 6: Gantt chart.

Such diagrams give detailed information about the simulation run and answer client's question in a very short time.

There are other diagrams and reporting tools available, which are even more meaningful and faster to grasp than these shown above. It is important then to select those, which suit one's style and experience. It would be best to enclose such a report tool into a module in the simulation software and just insert it to a simulation model when required.

If the client is fully informed about what happened in the simulation model then he can confidently draw valid conclusions from it. And this certainly increases his appreciation for this simulation technique.

7. GO ABOVE LIMITATIONS

I assume that every simulation tool is good in at least two of these features mentioned in this paper: providing parameters for playing with the model and full result set. But what if it is really hard to get good animation in your simulation tool? What if automatic building a model is simply out of question?

In that case it is the features that stand out that must be taken advantage of. Good simulation skills, good sense of what is really important in the simulation project will always help to prove the concept to the client.

But in the long run there is a higher chance of success of a simulation model with good animation against a model with poor one. 3D CAD tools are already commonly used, and the decision makers are used to such graphics. Thus, if your simulation environment allows only 2D then it will probably not make the expected impression on your client. Upgrading your simulation software to match the decent ones on the market, would be here well recommended.

Also a simulation tool with limited model generation capabilities (i.e. lack of detailed programming script) will be losing against a tool where every single operation in the model can be triggered from the code.

There is really no need to say that the simulation does not allow for something or to complain that the client wants too much. The simulation is really a powerful tool, one just has to open his mind and seek for solutions to the client's questions. Maybe it is just the simulation software that is not capable anymore, and needs to be replaced.

All the points mentioned in this article make the client in favor of using simulation again in the near future. This, in turn, helps to spread a good word about simulation.

REFERENCES

Barjis, J., Shishkov, B., 2001. UML based business systems modeling and simulation. *Proceedings of the Fourth International Eurosim Congress*, pp. 1-6. June 26-29, Delft (The Netherlands).

Fowler, M., Scott, K., 1999. *UML Distilled: Applying the Standard Object Modeling Language*. London, Adddison Wesley Longman Inc.

Heavey, C., McNally, P., 2004. Developing simulation as a desktop resource. *Int. J. Computer Integrated Manufacturing*, Vol. 17, 437-450.

Law, A.M., Kelton, W.D., 2000. *Simulation Modeling and Analysis*. 3rd ed. New York: McGraw-Hill.

Law, A., 2005. How to Build Valid and Credible Simulation Models. *Proceedings of Winter Simulation Conference*, pp. 24-32. Dec 4-7, Orlando, (FL, USA).

Stec, M., 2006. Fast Creation of Realistic and Efficient Free Path Network within a Simulation Model of a Shop Floor and a Supply Chain. *Proceedings of Winter Simulation Conference*, pp. 1531-1536. Dec 3-6, Monterey (CA, USA).

AUTHORS BIOGRAPHY

Michal Stec is a self employed simulation engineer, and the owner of SimulateFirst, a simulation company in Freiburg, Germany. He has received M. Eng. in Informatics from Wroclaw University of Technology, Poland and M. Sc. Eng in Industrial Engineering from National University of Ireland in Galway. He had worked for 6 years as a consultant for IT (Accenture) and discrete simulation (Simulation und Automation Technologies AG), managing successfully a number of simulation projects for known brands. In 2006 he set up SimulateFirst (www.simulatefirst.com), which offers simulation services and which meets clients' satisfaction in numerous applications in production, construction and logistics. He participates actively in simulation conferences; he published and presented articles at Winter Simulation Conference (2006) and at ASIM Symposium "Simulation in Production and Logistics" in Kassel (2006).