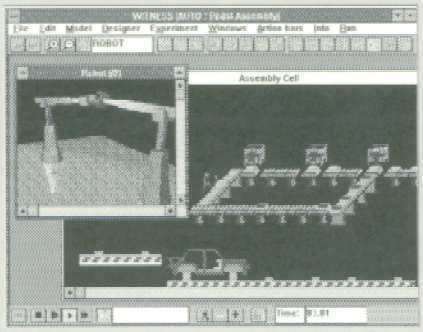
MANUFACTURING SYSTEMS

Insights via interactivity

The right simulation package furthers corporate goals

Because of a basic commonality among commercially available simulation packages, engineers tend to select software based on their personal preferences vis-à-vis certain "bell-and-whistle" type features. Cost also is a consideration.

Corporate management, however, tends not to care so much about the bells and whistles, and are able to look beyond the initial cost of the software. Its focus is on finding solutions or looking at the total life-cycle costs of using the software. The life cycle of simulation software involves not only model building, but also the iterative process of performing "What-if?" analyses to arrive at the most optimal solutions. Another important purpose of the simulation is management presentations.



(Photo courtesy Lanner Group)

Management should be aware that there is a basic distinction between simulation packages based on a "smart-parts, compiled-software method" versus the "visually interactive, interpretive" method that can make a real difference in the benefits gained from the use of simulation. It is also important to note that while most packages today can appear to be interactive, when the presentation layer is stripped away, some are revealed to remain compiled systems at their core.

Simulation lets users create models of real-world processes that are too complex to be analyzed by spreadsheets or flow charts. A model is a logical description of how a system or process behaves. Instead of interacting with the real system, a model is created that corresponds to it in certain key aspects.

With the "smart-parts, compiled-software method," the system logic is embedded in the work flowing through the model, rather than at the locations where the logic acts upon the work. This practice is a historical anachronism that goes back to when manual simulations were accomplished using students who stood in place of the work and carried with them the process instructions. The first computer simulations mimicked this state of affairs. Since then, developers have realized it is better to attach the logic directly to objects in the model. There are several reasons for this.

Model building

In smart-parts-based simulation software, the user sometimes is forced to change the logic of the "real system"—the actual process being modeled—so that, within the simulation, all logic is attached to either parts or entities. Within real systems, logical decisions actually occur at locations (operators or machines), not within parts or entities.

Changing the orientation of the real system logic to fit the smart-parts software model takes more engineering than would otherwise be the case. Hence, there are added costs when building models using the smart-parts method.

Simulation software that is compiled has another limitation. Once all the logic or decisions of the real system are attached to the parts or entities in the model, the user then must specify an exact run time and compile the entire code for the model. The modeler cannot interact with the model because it must complete running its pre-specified run time.

Since the modeler cannot interact with the model, the only way to correct modeling errors is to first run the compiled model for the pre-specified run time and watch the model run. Once the results have been interpreted, changes can be made to the model, and it must be recompiled and run again.

Simulation software based on smart-parts typically requires the user to learn a new modeling language particular to that software vendor. Typically, there is less reliance on visual objects in the model and more reliance on the particular modeling language. Some engineers prefer this system of job security, but corporate management must bear this additional cost of training and re-training engineers.

Visually interactive software allows users to choose not to attach all the logic or decisions of the real system to smart parts. Instead, the real-system logic can be attached directly to any object in the model. Correcting modeling errors also is more straight-forward. Since compiling is not needed, modeling errors can be corrected on-the-fly, while the model is running. A pre-specified run time is not required.

Visual status changes that occur as the model is running help identify possible modeling errors and the model can be changed immediately. In fact, when an error is detected, corrections can be recommended by the software itself. The reiterative error-correction process is eliminated.

"What-if?" analysis

To perform "What-if?" analyses, it's important to be able to make changes quickly. Otherwise, engineering costs will rise. Here again, the reiterative process of making changes and error corrections in the model can make use of smart-parts software more expensive. For the same reason, compiled software is not well-suited for use in giving presentations, since presentations by their nature must be interactive.

Visually interactive software lends itself much more readily to performing "What-if?" analyses. It also is better for making presentations. Because of its interactive nature, the demonstrator can "drill-down" to the details or instantaneously answer questions by changing model parameters. This benefit is not to be discounted. By being able to visually demonstrate the consequences of possible courses of action, the decision process becomes much more efficient.

Companies use computer simulation to increase productivity and compete more effectively. Interactive model control is a must in three areas: model building, "What-if?" analysis, and management or marketing presentations, if management is to realize the full benefits of simulation.—*Otto Thav, applications engineer, Lanner Group*