

# INTERACTIVE REFINEMENT OF A MATERIAL FLOW SIMULATION MODEL BY COMPARING MULTIPLE SIMULATION RUNS IN ONE 3D ENVIRONMENT

Matthias Fischer<sup>†</sup>  
Christoph Laroque<sup>\*</sup>  
Daniel Huber<sup>\*</sup>  
Jens Krokowski<sup>†</sup>  
Bengt Mueck<sup>‡</sup>  
Michael Kortenjan<sup>†</sup>  
Mark Aufenanger<sup>\*</sup>  
Wilhelm Dangelmaier<sup>\*</sup>

<sup>†</sup>Algorithms and Complexity  
Heinz Nixdorf Institute  
University of Paderborn  
Fuerstenallee 11, 33102 Paderborn  
Germany

<sup>\*</sup>Business Computing, esp. CIM  
Heinz Nixdorf Institute  
University of Paderborn  
Fuerstenallee 11, 33102 Paderborn  
Germany

<sup>‡</sup>Department MuT  
University of Applied Science  
Osnabrück  
Am Wall Sued 16, 49808 Lingen  
Germany

## ABSTRACT

The validation of material flow models as well as the selection of promising strategies for the generation of a successful experiment plan is a time-consuming process. A new approach is presented, which supports the simulation expert in his working process by giving him the opportunity to modify the simulated simulation run and afterwards compare the effects of his modification with the original setting, online and in one user interface, implemented by switching the visualizations between the simulation runs or opening up to 5 parallel 3D windows. The method developed therefore clones existing simulation runs online and allows the navigation within these existing simulation runs. The method has been implemented and is validated by a test model, which describes in detail the new working process of a modeler. New research questions are derived from this work, which will define following working steps.

## 1 INTRODUCTION

Today's state of art simulation tools, such as Flexsim, Automod, eM-PLANT, SIMUL8, Quest3D, and others, provide 3D visualization of the simulation models. Modeling and simulation can be done completely in 3D. Even during simulation, extending parameters and changing model logic can be done "on the fly". The user can manipulate the simulation model during the simulation run and an interactive analysis of the simulation is offered. 3D visualization offers benefits in intuitive process understanding and result communication (Kamat and Martinez 2000).

Recent, results from surveying simulation practitioners from industry who have used 2D or 3D software ap-

plications for discrete-event simulation show that many remain cautious, guarding against possible exaggeration of the claimed "3D benefits" (Akpan and Brooks 2005a).

The analysis of several sequential simulations remains difficult because of the missing direct comparison. Have the changes really enhanced the performance of the system? Because the unmodified system does not exist, the user has no possibility to compare the modified system with the unmodified system.

Our approach is to run *multiple similar simulations* in parallel. The simulation runs differ in the distinct parameter values of the model. Because of distinct parameter values, the simulation runs might show distinct behavior, e.g., they differ in the number of packets that are moving across a conveyor belt.

Executing the simulation in parallel is not enough: a small set of the simulation runs (in our implemented system five) can be directly seen in the 3D visualization simultaneously on one screen. A direct comparison is possible and the user can analyze the differences.

Further interactive modifications can be done by a "pausing-cloning-modification" process:

- first the user pauses all simulations,
- afterwards a single simulation is selected for cloning,
- after cloning, the user modifies parameters of the cloned simulation run.

After the modification of the simulation run, all simulations proceed and both, the unmodified and the modified, i.e., the cloned simulation, are rendered simultaneously, so that the user can see the differences in two separate 3D windows on the same screen.

In Section 2 we discuss related work in simulations with 3D visualizations and parallel and distributed simu-

lation. In Section 3 we present a conceptual overview of the system. We present the architecture and utilization of the system in Section 4. We discuss some derived research questions in Section 5 and conclude our ideas in Section 6.

## 2 RELATED WORK

Our simulation system consists of a walkthrough system that renders multiple simulation runs on a single screen. The user can clone a single simulation run, modify parameters, and observe the 3D visualization of the cloned and modified simulation run. Therefore, first we review 3D visualization of simulation with 3D support, second we briefly discuss distributed simulations and cloning of simulations.

### 2.1 Interactive Simulations with 3D Visualization

The 3D visualization of simulated processes has successfully found its way into discrete simulation technology for the last decade (Barnes 1997); before the discrete simulators rendered the model in a 2D fashion with icons and lines (Hlupic 2000). Typical arguments for 3D visualization of a simulation modes are a better understanding of problem by all stakeholders and the easier identification of errors (Kamat and Martinez 2000), e.g., the detection of accumulations of packets on vertically arranged conveyor belts.

Most of today's discrete simulation systems (e.g., Quest, Automod, and others) already provide integrated 3D visualizations. In these systems, simulation and rendering running on the same machine. For performance reasons, simulation and rendering can run on different machines and are bidirectionally coupled with each other (Strassburger et al. 2005, Fischer et al. 2005). When coupling simulation with VR on different machines, the synchronization of both tools, i.e. the time advance in both tools, must be coordinated (Strassburger et al. 2005). Not only virtual reality is an application of 3D visualization, but also augmented reality based simulations (Behzadan and Kamat 2005).

Recent results of experimental studies that tested the impacts of Virtual Reality (VR) on Discrete-Event Simulation show that it is easier and faster to spot errors in 3D/VR model than in 2D (Akpan and Brooks 2005b). Despite the advantages of 3D visualization, many remain cautious (Akpan and Brooks 2005a) because of a long learning curve for 3D software.

We think the features of the 3D visualization are crucial to get a benefit. Our contribution in this paper is a new 3D visualization that supports the modeler by allowing comparison of settings before and after model modification.

### 2.2 Distributed Simulation and Cloning

To overcome the computation and memory limitations of single processor architectures, parallel and distributed execution of discrete event simulations is used to speedup the computation. One approach is the concurrent execution of a single simulation run in a multi-processor architecture. Another technique is to execute multiple runs of independent simulations in parallel on a set of concurrent CPUs. Every simulation run is executed from the beginning to the end on the same CPU. The single independent simulation results are collected for correct and statistically significant analysis. (Banks et al. 1996, Fujimoto 1990, Kalyan 2006).

We use a loosely coupled PC cluster for our simulation and visualization system. Rendering and simulation runs are executed simultaneously on concurrent CPUs. The user switches between the simulation runs in order to observe the progress and investigate the differences.

Cloning of parallel simulations is designed to analyze alternative scenarios of a distributed simulation concurrently within the same execution session. The goal is to reduce the execution time for evaluating different scenarios by avoiding repeated computation. A simulation is cloned on an event for a particular set of logical processes, creating new clones (Hybinette and Fujimoto 2001). Just-in-time cloning mechanisms (Hybinette 2004) avoid unnecessary cloning of logical processes. Schulze et al. (1999) introduced a cloning approach based on HLA to extend the flexibility of system composition during runtime, and proposed the cloning of federates at runtime. Forecast functionality is provided by their approach via the parallel scenario management of different time axes. Chen et al. (2005) improves the cloning of HLA-based distributed simulations by cloning mechanisms and infrastructures that ensure state consistency and support user transparency and reusability of federate codes.

Our system provides cloning of simulations as means for running multiple similar simulations in parallel. However, unlike (Hybinette 2004), our goal is not to share computations. Some of the cloned simulations are modified by manipulation of parameters. The modified simulations are rendered and visualized in parallel in a 3D walkthrough system. Our cloning implementation uses ideas of the approach of Schulze et al. (1999). We store all relevant elements of the simulation model, such as object instances and other internal states of the model.

## 3 CONCEPT

Based on the fundamental idea of improving and advancing the workflow of a simulation expert in the area of the material flow simulation, the concept presented and its implementation uses an interactive material flow simulator with 3D-visualization not only for the representation of one single simulation run, but to show noticeable dif-

ferences and similarities between multiple simulation runs in one virtual environment.

The development process of a simulation model by a simulation expert can be enhanced if the interactive modifications and their consequences can be compared to the original setting during the continuous execution of the simulation run. Variants in the simulation model, as well as different control strategies can be derived from the original setting in one working step and without restarting the simulation run.

### 3.1 Overview

The main goal of this work is to enhance and accelerate the simulation expert's work in designing and validating a material flow simulation model. Moreover, the user will be enabled to design better model variants and control strategies based on the original setting. Therefore, in a first step, an interactive environment is needed which generally allows the modification the system modeled and its dynamics during its execution and follows the effects of which behavioral changes result restrictedly from the interaction and which were already defined in the original model, the user needs a possibility to compare the original simulation run with a configuration which includes his modifications. Thereby, this comparison allows judgement of the modifications aimed at the goal of model improvement.

We want to achieve This goal by the use of a combination of interactive modifications, the use of a cloning mechanism to create parallel simulation runs without restarting the simulation, and the representation and control in one single 3D-visualization system.

### 3.2 Workflow

The implementation of the ideas presented also leads to changes of the user's workflow in order to improve and accelerate his work. The resulting workflow and its consequences for the implementation of the system architecture is described in this paragraph. So how does it look?

The user develops and designs a first draft of a simulation model and simulates it in order to check the modeled behavior. Thereby, a 3D-visualization in a virtual environment helps him to easily recognize the system dynamics and compare them to the goals of the simulation study. The user has to validate the original simulation model in respect to the experimental frame or change the behavior of the modeled system. Normally, this leads directly to the loss of the original setting, so that the following dynamics of the simulation run cannot be divided into those which result directly from the modification and those which were already defined in the model. When using common simulation tools, the user has to stop the simulation, modify the original model and restart the

simulation, which leads to a time-consuming work process.

The process of an interactive modification is accelerated by a cloning mechanism (see Figure 1), which is started after a user-defined set of interactions. This mechanism clones the original simulation run with its current state and assigns the set of interactions only to the copy of the simulation run. This way, the original setting is saved and the consequences of the interactions can be compared directly and without restarting the simulation run.

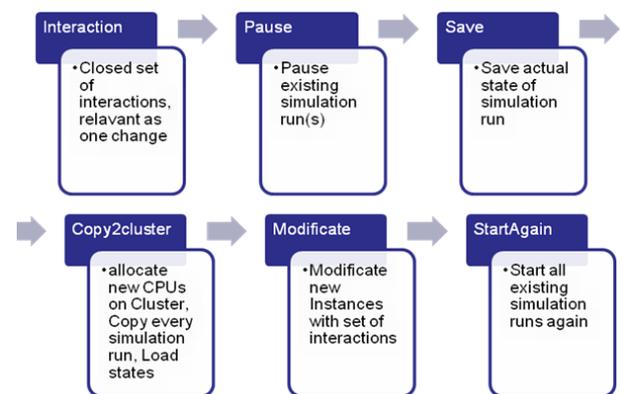


Figure 1: Cloning Process

By the additional use of a visualization system, which allows switching between or tile the 3D-visualization of each existing simulation run (more than one set of interactions clearly leads to several simulation runs), the direct comparison of different simulation runs is allowed.

As a consequence of this workflow, the need for a platform for several simulation runs and their visualizations arises which allows the execution of 10 or more simulation runs, even if the simulation model itself is very complex. During the implementation, a system architecture was to be developed as first step, which supports this working process of a simulation expert. An effective navigation in a big set of simulation runs is difficult, but can be improved, by designing a visualization method, which allows the viewing and control of the existing simulation runs. In this approach, an “experiment tree” was developed (see Figure 2) , which covers this method and supports the user in finding the right simulation runs for a meaningful comparison.

The computational power needed cannot be restricted to the simulation runs exclusively because navigation within the experiment tree leads to the switching of the visualizations, or, in case of multiple rendering windows, to the visualization of several simulation runs in one user interface. Therefore, the existing computational power has to be balanced between the simulation runs and their rendering as well as with the rendering itself. During the implementation of the system architecture, load-balancing problems are also solved.

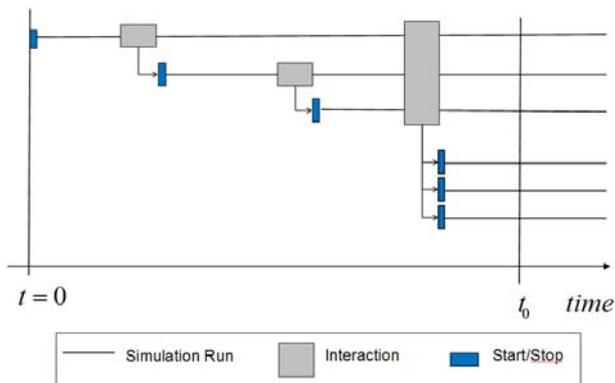


Figure 2: Experiment Tree

### 3.3 Advantages

By pursuing this new user workflow, the simulation expert is able to improve his understanding of the process and has the opportunity to compare and, therefore, directly judge the consequences of his modifications in a running simulation model. Using this new method of analysis, he is able to evaluate if his modifications indeed lead to a better performance of the simulation model or one of its variants. Different control strategies can be compared in one view, so that only those, which promise maximum success are to be ensured through simulation experiments. Validation and improvements during the modeling phase of a simulation experiment can be made faster, safer and, as a result, with less time and cost.

## 4 UTILIZATION IN MODEL ANALYSIS

In this section, the workflow in the system designed is demonstrated. Starting with the description of a small model, the usage of the system and the techniques to compare different system configurations are presented.

### 4.1 The Test Model

The model consists of two workers doing quality control on three different product types. The quality tests demand manual work at the benches B1-4. Worker W1 works at benches B1 and B2; worker W2 works at benches B3 and B4. Product P2 is an uncommon product, but when pieces arrive at B1, they are of top priority and have to be tested immediately. P3 is incoming at a constant rate, but it is not very important. When the input rate of P2 is normal, W1 can test products P1 and P2 alone. To absorb peaks in the input rate of P2, a distributor D can divide P1 between B2 and B3. If so, W2 has to test P1 on bench B3 next to his normal work.

In the following paragraphs, this model is used in an experiment to find a strategy for distributing P1 between the

two workers in case of a peak in the input rate of P2. It is assumed that there is a delay time until the realization of the peak and the start of distribution of P1. So there will be a queue of P1 in front of B2 before the distribution begins. The distribution should assure that there are not many waiting pieces of P1 and W2 is only involved in testing P1 at a minimal level.

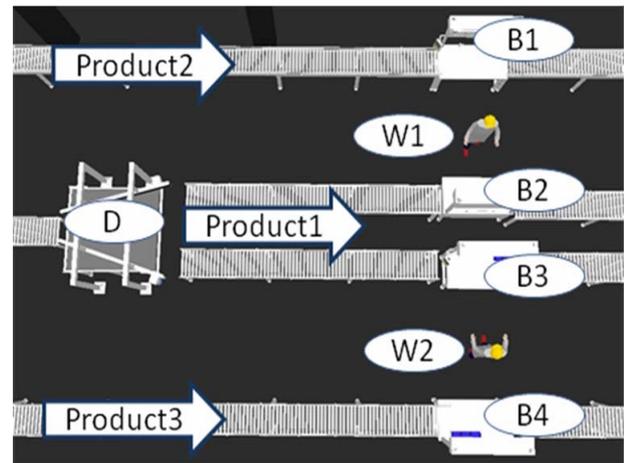


Figure 3: Test Model Overview

### 4.2 Step 1: Creating new Simulations with varying Parameterization by Cloning

Assume the user wants to compare systems with distribution rates between B2 and B3 by 1:0 (only B2), 1:2 and 1:1 (equal distribution). The rate of 1:0 is applied in the original system, thus two new simulations have to be created. To create a new clone, the user has to:

1. Pause the simulation
2. Apply the parameter changes on the simulation entities. In this case setting the distribution rate to 1:2. (Clone 1 in Figure 4)
3. Initiate the cloning process
4. Repeat with 2. to set the distribution rate to 1:1. (Clone 2 in Figure 4)
5. Unpause the simulations

The actual cloning in 3. is performed by:

1. Saving the unchanged simulation
2. Creating a new simulation kernel and renderer
3. Loading the saved simulation without canceling the original one
4. Applying the parameter changes to the cloned simulation

### 4.3 Step 2: Comparing the System Configurations

After cloning, three simulation runs exist. Each one is visualized in a window (Figure 4). The model performance can be analyzed by viewing all windows tiled next to

each other. In this example, it can be seen that the number of pieces waiting in front of B2 increases when the distribution rate is 1:0. The rate of 1:2 over compensates the effect and 1:1 generates a equal queue length. Thus setting the rate to 1:1 in case of a peak seems to be a good strategy.

Comparing three different systems is not difficult. But the number of simulations can increase rapidly when analyzing several strategies simultaneously. One scenario would be to find an optimal distribution rate if there is an option to test only every second or third piece. This would result in  $3(\text{distribution rate}) \times 3(\text{test volume}) = 9$  simulations.

#### 4.4 Step 3: Navigating the Experiment Tree

To analyze an experiment with more than a hand full of simulations, more efficient techniques are necessary. To organize all simulations created, trace changes, or stop

single simulation runs, an *Experiment Tree* was created (Figure 4, top left). Every simulation run is a line extending over the time axis. When clones are generated, new lines are created indicating which simulation is the origin. When picking one line, the corresponding simulation is visualized in the main window. If multiple lines are picked, multiple simulations are visualized, one in the main window, the others in the smaller ones. .

### 5 SYSTEM ARCHITECTURE

In this section, the system architecture capable of computing and visualizing parallel replications of simulations is presented.

The number of simulations created by cloning can increase rapidly, thus implementation on a simple computer is not feasible and the system has to be built to run on a computer cluster. The system architecture is shown in Figure 5.

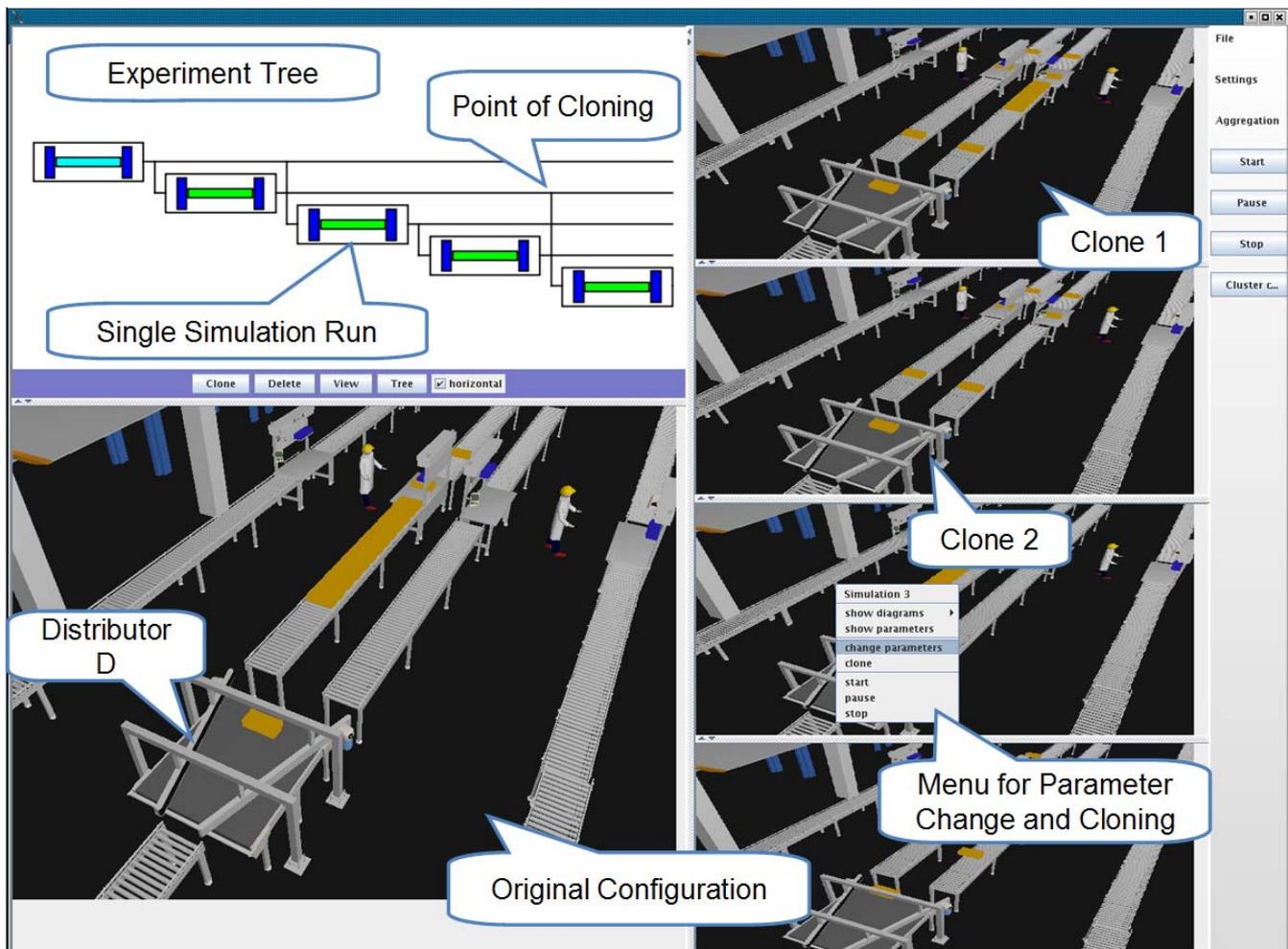


Figure 4: Graphical User Interface

A number of  $n$  nodes (SR) render the static objects of a simulation model. These could be conveyor belts, machines, etc. Because static objects are the same in all cloned simulations and the view frustum in every visualization window is the same, the static objects have to be rendered only once, no matter how many simulations are compared. To allow efficient parallel rendering of the static objects, a load balancing is done between the (SR) nodes. Therefore the view frustum is tiled in rectangles with an equal amount of polygons.

The dynamic objects, like tokens or forklifts, can differ in every cloned simulation. With no difference in dynamic objects, two simulations are exact copies. Thus one image of the dynamic objects per simulation has to be generated. One node (D) per simulation is allocated for rendering the dynamic objects (DR) as well as computing the simulation (Sim). For computing the simulation, the simulator d<sup>3</sup>FACT is used (Dangelmaier et al. 2005). For each clone created, one new node (D) has to be allocated. All rendered images are composed into one final image on the Mother Node. The image composition in the Mother Node first composes the static scene by joining the images send by SRs. Secondly, the image of all visible dynamic objects of one simulation is added. By using the Z-Buffer, occlusion of dynamic objects by static ones is achieved.

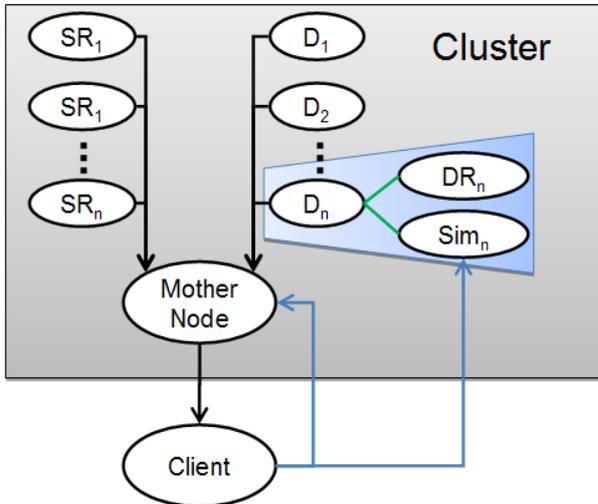


Figure 5: System Architecture

This final image is sent via Ethernet to a client computer exterior to the cluster. The in-cluster communication is transferred via Infiniband for higher performance. Because the Mother Node composes the image shown in the client, the client has to send information about the view frustum and the simulations to be visualized. For cloning and simulation data acquisition, the client can communicate directly with the simulations on the nodes (D).

To allow efficient comparison of simulations, the simulation times of all clones have to be synchronized. But because there is no interaction between simulations,

the accuracy of this synchronization does not have to be very high. Synchronization is done by giving the simulations a time interval of simulation time for computation and waiting until all simulations have reached the end of the interval. After that, a new interval is sent to the simulations.

## 6 DERIVED RESEARCH QUESTIONS

During the implementation of the ideas presented, several further ideas by strictly following the user's demands arise. These ideas define new research areas still to be realized. The most important are described below.

### 6.1 Aggregation of Multiple Simulation Runs in a Single Virtual Scene

By working with the system architecture designed, the comparison of different simulation runs gets more and more complex when the number of interactions increases. After some interactions, either the user gets more and more confused by the often necessary switches, or the number of parallel windows is so high that the differences within the visualized simulation runs are not obvious any more.

As a solution, the presentation of several different simulation runs in one common virtual scene is imagined. Special work is still to be done to face those data, which is to be emphasized for an easy and fast analysis. As a first step, a scheme is to be defined which covers possible differences between these simulation runs, e.g. static differences (number of machines) or dynamic differences (position of a forklift or a package on the conveyor). The scheme designed also has to cover possibilities for the visualization of different internal states, e.g., the current state of a machine or its workload. For each element, an adequate type of visualization is to be identified. For example the question arises, how to visualize the aggregation of the forklifts' routes.

### 6.2 Visualization of Stochastic Processes

The material flow simulation is furthermore a stochastic process, which directly leads to the insight that the origin simulation, its dynamic behavior, and its visualization is not necessarily representative for the system mapped. The origin setting therefore should not consist of a single simulation run, but should cover a configuration of distributed simulation experiments, where every single simulation run just differs from the others in the start values of the stochastic parameters. The cloning process gets more complex this way, because several simulation runs are to be paused, saved and distributed within the computing cluster. This complicates the balancing problems mentioned and increases the computing power necessary.

Nevertheless, the number of simulation runs increase in a more relevant way with interactions: in the current system setting, two sets of interactions lead to up to 4 simulation runs. Starting with a configuration of 10 distributed simulation runs, the same interactions would lead to 40 distributed simulation runs, which sooner or later will cover all available computing power in the cluster. Therefore, new methods must be identified by which the number of simulation runs can be reduced by selecting only relevant simulation runs and thereby thinning out the entire experiment tree, which itself has to be improved for an efficient navigation within such a complex set of simulation runs.

But the intended solution also offers additional possibilities for an efficient analysis by the user. By regarding the entire configuration of more or less 10 simulation runs, the system can offer the user not only a specific parameter value, but also an aggregated range of existing values within this configuration additionally enriched by statistical analyses, e.g. average, median, standard deviation, etc. This allows the correct classification of the visualized scene and gives more detailed information about the system behavior.

## 7 CONCLUSION

In order to improve the workflow of a simulation expert dealing with the material flow simulation, a new approach has been introduced which allows the direct comparison of simulation runs in one user environment. By cloning a simulation run after a set of interactions, the validation and control process of the modeler can be accelerated because the effect of the interaction can be judged directly. The system therefore clones the running simulation and refines just the clone with the set of interactions. The method has been implemented and validated by a test model, which leads to new research questions. After a large number of interactions, the switching between the resulting simulation runs gets more and more complex. New methods need to be found to visualize different simulation runs in one window. Furthermore, the starting simulation run is not necessarily representative for the mapped system. To avoid this source of error, the system architecture is to be enlarged, so that the starting simulation run is broadened to a configuration consisting of round about 10 simulation runs.

## REFERENCES

- Akpan, J. I., and R. J. Brooks. 2005a. Practitioners' perception of the impacts of virtual reality on discrete-event simulation, In *Proc. of the 2005 Winter Simulation Conference*, 1976-1984.
- Akpan, J. I., and R. J. Brooks. 2005b. Experimental investigation of the impacts of virtual reality on discrete-event simulation, In *Proc. of the 2005 Winter Simulation Conference*, 1968-1975.
- Banks, J., J. S. Carson II, and B. L. Nelson. 1996. *Discrete-Event System Simulation*. Upper Saddle River, N.J.: Prentice Hall.
- Barnes, M.R. 1997. An introduction to QUEST. In *Proc. of the 1997 Winter Simulation Conference*, 619--623.
- Behzadan, A.H., and V. R. Kamat. 2005. Visualization of construction graphics in outdoor augmented reality. In *Proc. of the 2005 Winter Simulation Conference*, 1914 - 1920.
- Chen, D., S. J. Turner, W. Cai, B. P. Gan, and M. Y. H. Low. 2005. Algorithms for HLA-based distributed simulation cloning. *ACM Transactions on Modeling and Computer Simulation* 15(4): 316 – 345.
- Dangelmaier, W., D. Huber, C. Laroque, B. Mueck 2005. D<sup>3</sup>FACT insight – An immersive material flow simulator with multi-user support. In *Proc. of the 2005 Summer Computer Simulation Conference*, 239-242
- Fischer, M., B. Mueck, K. Mahajan, M. Kortzenjan, C. Laroque, and W. Dangelmaier. 2005. Multi-user support and motion planning of humans and humans driven vehicles in interactive 3D material flow simulations. In *Proc. of the 2005 Winter Simulation Conference*, 1921-1930.
- Fujimoto, R. M. 1990. Parallel Discrete Event Simulation. *Communications of the ACM*, 33(10), 30-53.
- Hlupic, V. 2000. Simulation software: An Operational Research Society survey of academic and industrial users. In *Proc. of the 2000 Winter Simulation Conference*, 1676 – 1683.
- Hybinette, M. 2004. Just-in-time cloning. In *Proc. of the 18th Workshop on Parallel and Distributed Simulation*, 45–51.
- Hybinette, M., and R. M. Fujimoto. 2001. Cloning parallel simulations. *ACM Transactions on Modeling and Computer Simulation* 11(4):378–407.
- Kalyan S. P. 2006. Parallel and distributed simulation: traditional techniques and recent advances. In *Proc. of the 2006 Winter Simulation Conference*, 84-95.
- Kamat, V. R., and J. C. Martinez. 2000. 3D Visualization of simulated construction operations, In *Proc. of the 2000 Winter Simulation Conference*, 1933-1937.
- Schulze, T., S. Straßburger, and U. Klein. 1999. Online data processing in simulation models: New approaches and possibilities through HLA. In *Proc. of the 1999 Winter Simulation Conference*, 1602–1609.
- Strassburger, S., T. Schulze, M. Lemessi, and G.D. Rehn. 2005. Temporally parallel coupling of discrete simulation systems with virtual reality systems. In *Proc. of the 2005 Winter Simulation Conference*, 1949-1957.