Proceedings of the 2008 Winter Simulation Conference
S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler eds.

AGGREGATED 3D-VISUALIZATION OF A DISTRIBUTED SIMULATION EXPERIMENT OF A QUEUING SYSTEM

Wilhelm Dangelmaier
Business Computing, esp. CIM
Heinz Nixdorf Institute
University of Paderborn
Fürstenallee 11
33102 Paderborn, GERMANY

Matthias Fischer
Algorithms and Complexity
Heinz Nixdorf Institute
University of Paderborn
Fürstenallee 11
33102 Paderborn, GERMANY

Daniel Huber
Christoph Laroque
Business Computing, esp. CIM
Heinz Nixdorf Institute
University of Paderborn
Fürstenallee 11
33102 Paderborn, GERMANY

Tim Süß
Algorithms and Complexity
Heinz Nixdorf Institute
University of Paderborn
Fürstenallee 11
33102 Paderborn, GERMANY

ABSTRACT

The paper describes an approach for an aggregated animation of a simulation experiment in an interactive 3D environment, visualizing multiple, distributed simulation runs. Although the general approach of a 3-dimensional visualization of material flow simulation helps to understand the dynamic behavior of a system better as well as faster, it remains unclear, how typical the animated simulation represents the model, if there is a stochastic influence for even some parameters. By the integrated visualization of multiple distributed simulation runs, this uncertainty can be solved, which will be shown in this paper for a typical simulation study of a queuing system.

1 INTRODUCTION

Briefer product lifecycles, customer-oriented production and raising variants of the manufactured goods are representative for today's industries. In order to manufacture effective regarding time and costs, the digitalization of product and process planning is of growing importance. New products are designed and optimized nearly completely digital. The advantages are obviously lower developing cost and times and thereby a better competitive position for the company. But the ongoing digitalization is beyond product design and also affects all relevant areas of process planning (VDI 4499, 2007). The affected companies try to realize the vision of a "digital factory".

One well-established method for planning, safeguarding and improvement of production processes is the material flow simulation (Law and Kelton 2000), which allows to create and improve structural models and interpret the results of their dynamic execution. During the last years, the 3D-visualization of material flow simulations has become of growing importance, because it helps the user to understand the underlying behavior of the modeled system easier and faster. Moreover, the 3D-view of the simulation model offers help in communication and discussion of decisions with non-simulation experts.

Stochastic influence of parameters results in the problem that an animation of a simulation run cannot necessarily be regarded as typical of the behavior, because it might also be possible to view the best or the worst case. In the classic experimentation, this leads to the repetition of a simulation run, thus creating a simulation experiment, in order to allow the judgment of each simulation run in context to a set of simulation runs.

If these multiple simulation runs are distributed on a PC-cluster and calculated in parallel, each simulation run can be used for visualization or even multiple simulation runs of one experiment can be visualized in one 3D-environment, in order to allow the direct comparison within the simulation experiment (Fischer et al. 2007). Tests

978-1-4244-2708-6/08/$25.00 ©2008 IEEE

show, that this only seems appropriate for a visualization of less than 5 simulation runs in parallel.

Derived from this insight, the general approach presented in this paper is to visualize not only one simulation run, but animate all simulation runs in one 3D-visualization interface by using percentaged transparency for each dynamic object of each simulation run within the simulation experiment. Thereby it will be possible to visualize the stochastic derivations of the simulation experiment in one user interface, which, in additional step, will allow the direct judgment of possible user interactions during the execution of a simulation in the whole stochastic width. Moreover, the identification of extreme simulation runs can be done faster and the direct analysis of reasons can be allowed.

In a first step, this paper will in brief present the necessary state-of-the-art in this research area, which includes the prefabricated work of the authors. Then the general approach will be discussed in more detail, based on a typical example of a queuing system. The technical implementation is presented and the paper closes with an outlook on further research, in order to improve existing and create new ideas.

2 RELATED WORK

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); typically, the discrete simulators rendered the model in a 2D fashion with icons and lines (Hlupic 2000). Typical arguments for 3D visualization of simulation models are a better understanding of the regarded 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.

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 an 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.

Most of today's discrete simulation systems (e.g., Quest, Automod, and others) already provide integrated 3D visualizations. In these systems, simulation and rendering are running on the same machine. For performance reasons, simulation and rendering can be executed on different machines and 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).

2.2 Experimentation With Stochastic Influences

Possible existing disturbances of a model element within a simulation model are typically modeled via random number variables, so that the appearance of a specific event can only be estimated by probabilities. These stochastic influences lead to the multiple repetition of a specific simulation configuration with changed initial parameter values for the stochastic variables, which determine the random number streams for each parameter (Law and Kelton 2000). This set of similar simulation runs is regarded as a simulation experiment. In the presented approach, this simulation experiment is distributed on a PC-cluster, in order to allow the parallel aggregation of data out of each simulation run in one user interface with a 3D-visualization.

3 D³FACT INSIGHT

3.1 Basics

Besides the support of concurrent collaboration of multiple simulation experts on one simulation model, especially the improvement of modeling and simulation by the execution in a 3-dimensional environment was objective of the development of the material flow simulator d³FACT insight. The user himself is integrated in the highest possible way. Moreover, new procedures within the project have led to advances in the daily modeling work.

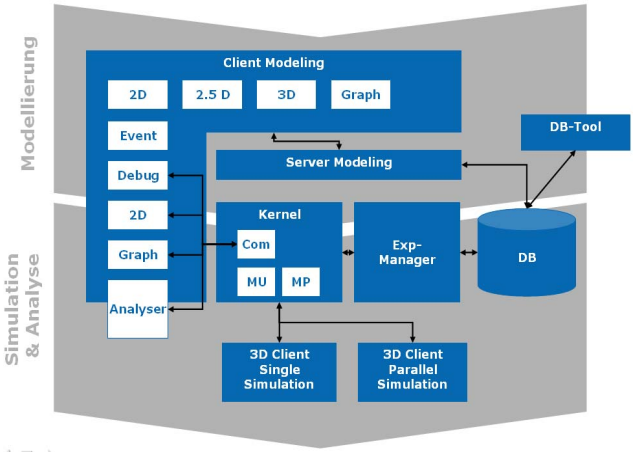

Figure 1: Modules of d³FACT insight

The simulation model can be presented, visualized and executed in different visualization modules. It can also be easily distributed on a cluster of computer systems, in or-

der to enable a distributed simulation of multiple simulation runs of one simulation experiment. The separation in modules follows functional aspects, and was made as described by Figure 1.

For each element of the simulation model, it is possible to connect a 3-dimensional representative from the simulation database, so that the generated scene can be displayed in the 3D-frontend. The 3D visualization front-end has been developed with new rendering methods, so that even large and complex scenes can be displayed during a real-time simulation run. The simulation model is to be computed in the simulation kernel, which is a discrete, event-driven simulator. Besides the internal simulation time, the simulation kernel synchronizes the real-time of the different clients.

d³FACT insight bases on the general concept of a simulation-based planning and control and is enriched continuously. During the implementation, the system is used in practice constantly, to fulfill even new demands, which are rising from the daily work of simulation experts.

3.2 Visualization of Distributed Simulation Runs

In (Fischer et al. 2007), the authors presented a basic work for this approach. The basic objective was to enhance and accelerate the simulation expert's work in designing and validating a material flow simulation model. Moreover, the user is enabled to design better model variants and control strategies, based on the original setting. Therefore, in a first step, an interactive environment generally allows the modification of the modeled system, its dynamic behavior and keeps the effects of this modification. In order to allow any analysis of which behavioral changes result restrictedly from the interaction and which were already defined in the original model, the user gets a possibility to compare the original simulation run with a second configuration, which includes his modifications. Thereby, this comparison allows the judgment of the modifications aimed at the goal of model improvement. This is achieved by using a combination of interactive modifications, a cloning mechanism to create parallel simulation runs without restarting the simulation and the representation and control in one single 3D-visualization system (see Figure 2).

Although with this system, it is already possible to get a better insight in the stochastic width of a simulation experiment, the direct comparison of simulation runs is accomplished by split-screen visualization, which can be used efficiently only for a few parallel simulation runs. The visualization of more than 10 simulation runs is hard because of the limited display space. Figure 3 shows a situation of sixteen simulation runs. It is hard to detect details in a single simulation run because of the small rendering windows of each simulation run. To improve the system, a technique is needed to visualize multiple simulations in one user interface.

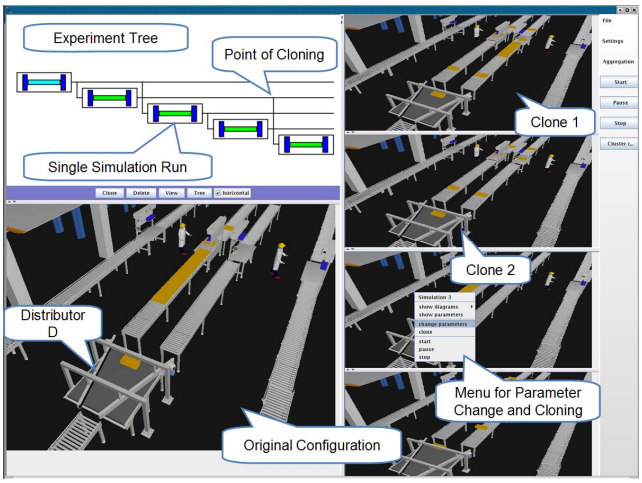

Figure 2: Visualization of multiple (here 4) simulation runs in one user interface (Fischer et al. 2007)

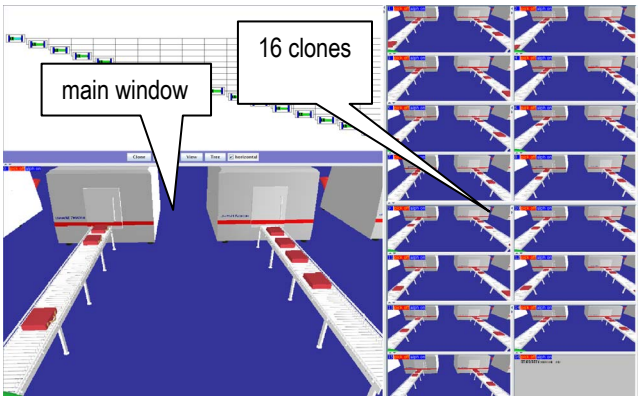

Figure 3: Sixteen simulation runs in one user interface.

4 CONCEPT

The methods presented in this paper visualizes stochastic influence of parameters in a 3D environment. Stochastic influence means starting several simulation runs with the same simulation model but with different seeds of random variables. In contrast to mathematical theory, in practice the stochastic width of an experiment is sometimes not calculated exactly because of the complexity of the simulation model. Rather empirical values are used ,e.g., an experiment of a material flow simulation is executed with a stochastic width of 10 to 20 simulation runs. In our experiments we use 8 or 16 simulation runs.

The problem is open to visualize the different simulation runs in a simulation software that features 3D visualization of the simulation model. Such a tool renders a single simulation run with the help of a virtual 3D model. Mathematical terms showing the stochastic influence of parameters are well known, e.g., average, variance, and so on. However the question remains how stochastic influence

can be visualized in a single 3D environment. For example, tokens moving across a conveyor belt may be located differently for different simulation runs.

This work presents visualization methods that should be a mean for the visualization of stochastic influence in a single 3D environment. Several simulation runs will be aggregated in a single 3D scene of the simulation model. The method aggregates the state of several simulation runs to a single 3D scene of the simulation model. Aggregation means an accumulative view of all simulation runs in a single 3D scene.

First, the example scenario used to demonstrate the techniques is described in Section 4.1. The following sections are used to demonstrate the technique in more detail and discuss drawbacks and opportunities.

4.1 Example Scenario

To show the benefit and the application of the designed system, two parallel executed replications of one model are used. Each model consists of 5 parallel M/M/1 queuing systems (Figure 4).

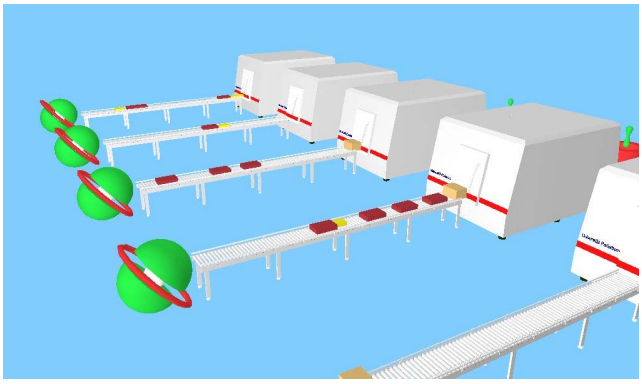

Figure 4: Example Model, 5 parallel M/M/1 Queues

Each queuing system consist of one source (green), one conveyor belt, one server (white cube) and one sink. The mean value of the interarrival time of the source is 20% higher than the service time of the server. According to queuing theory, this should result in a expected mean queue length of approx. 4.1. The conveyor belt was set to a length and conveying speed, so that the transport time is equal to the mean of the interarrival time. The token on the conveyor can only be transported in the direction of the server until they reach the physical end of the last waiting token, if there is one. So tokens pile up while waiting to get served. The source produces three different types of tokens, each with another length. The lengths of the tokens are chosen in such a way, that 18 of the smallest or 8 of the largest token fit on one belt. When a new token is produced by a source the type of this token is chosen according to a probability distribution. To achieve a non uniform behavior over all 5 parallel queuing systems, each random distribution, i.e., service time, interarrival time and token type choice, has another seed.

According to this model definition, the occupancy situation on each conveyor belt at one point of time is fairly different. Depending on the actual interarrival times and the service times the conveyor belts can contain more or less tokens. Additionally, depending on the type of tokens the total length of all tokens waiting in front of the server can be different.

If the simulation is analyzed in a virtual reality environment, where the observation time is not very long, wrong conclusions could be made recording to the optimal length of the conveyor belts. This is because the user only gets a snapshot of the model behavior. One snapshot could be, that the conveyor belts are quite full, another one, that they are empty.

If using aggregated visualization, as it is presented in this paper, the user gets several snapshots at one time and thus effects resulting from randomness are leveled out by a visual mean.

4.2 Aggregation of Uniform Tokens

Goal of the presented techniques is a simulation tool that features an automatic visual aggregation of stochastic influences. The aggregated data will be visualized in a single 3D scene. The data differs in their state depending on the simulation runs. In each simulation run, tokens are transported across a conveyor belt. The location and the number of transported tokens depend on stochastic influences. Some simulation runs get longer tailbacks and some of them shorter ones.

The goal is to generate a single 3D scene that gives the user an impression of the tailbacks of all simulation runs. Short and long tailbacks have to be aggregated in a single 3D scene. The approach in this paper uses transparency for the aggregation of the tokens. A token of each simulation run is rendered with small transparency. The transparency value is the fraction determined by the number of simulation runs. For instance, if the experiment consists of 10 or 20 simulation runs, the transparency value is 0,1 or 0,05 respectively (transparency value 1 means opaque). If in 10 of 20 simulation runs a token is located in the same place, the rendered aggregated tokens are an accumulation which is equivalent to a rendered single token with a transparency value 0,5.

Figure 5 shows decreasing transparencies of a token. Lowest transparency is 1/8, and highest is 1,0. Each token has additionally 1/8 transparency more than the preceding one. The following subsections show how the rendered images are interpreted.

Figure 5: Decreasing transparencies of a token.

4.2.1 Decrease of Transparency shows Stochastic Influence

The simulation runs of this section (Section 4.2) are restricted to tokens of equal size and color (red). The rendering technique results in images that show tokens with different transparency. Figure 6 shows such a situation:

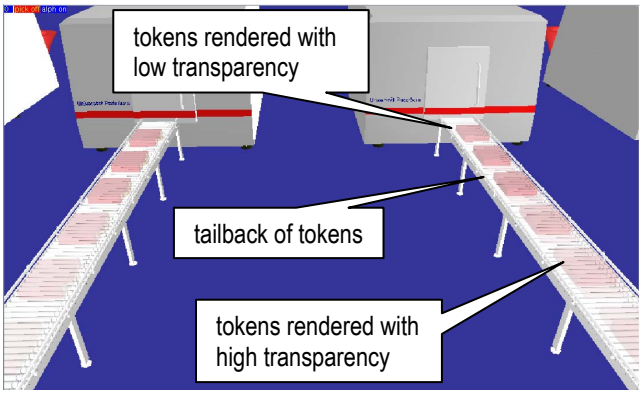

Figure 6: Tokens in front of the tailback are rendered with lower transparency as tokens at the end of the tailback.

Tokens in front of the conveyor belt are rendered with low transparency. Low transparency means that tokens of many simulation runs are located there. The tokens at the end of the conveyor belt are rendered with higher transparency. Higher transparency means that only few simulation runs locate tokens there. The conveyor belt shows a decrease of transparency from back of the conveyor belt to the front. This kind of visual aggregation means that many simulation runs have tokens in front, but only few simulation runs have tokens at the end of the conveyor belt. The decrease of transparency shows an increase of tokens. The aggregated 3D scene gives an impression of the stochastic influence of the simulation runs.

4.2.2 Tail End and the Space behind

The aggregation technique makes it possible to distinguish between states that have lots of simulation runs in common and states that have only a few simulation runs. This feature will be explained by means of the example shown in Figure 7.

Both conveyor belts show a tailback of tokens. The tail end of the left conveyor belt is after 2 tokens and the tail end of the right is after 3 tokens. The tokens are rendered by an aggregation of transparent tokens of all simulation runs. The front of the conveyor belt shows nearly opaque tokens. Opaque tokens are an indication that all simulation runs have a token there. The user can conclude that the tailback in all simulation runs is at least 2 tokens on the left conveyor belt and 3 tokens on the right conveyor belt.

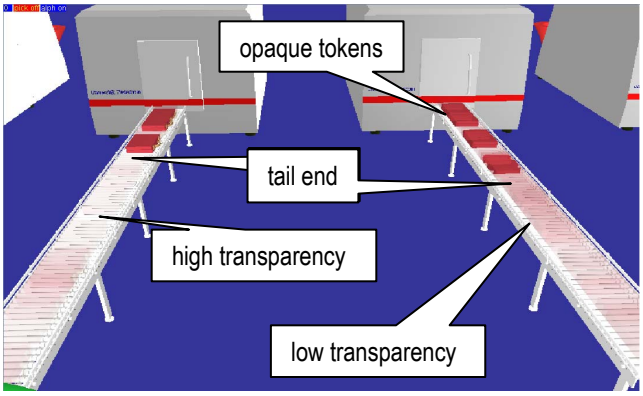

Figure 7: Both conveyor belts show tokens that pile up in all simulation runs. Some simulation runs have longer tailbacks.

Behind the tailback of the left conveyor belt is a blur of tokens rendered with high transparency. The right conveyor belt shows a mist of tokens with lower transparency. The transparency of tokens derives from a few simulation runs that have a longer tailback than the others. High transparency of the left tailback and lower transparency of the right tailback means that only a few simulation runs of the left conveyor belt and a little bit more simulation runs of the right conveyor belt have longer tailbacks. The user receives an impression of how much simulation runs contribute transparent tokens to the final image. However it is impossible to find out which simulation run contributes tokens.

4.2.3 Time Lags

Time lags between simulation runs can be a problem for the aggregation of data. If two simulation runs have a similar status, e.g., both conveyor belts transport the same tokens and the location of the tokens are nearly identical. However, the tokens are shifted against each other so that the tokens are slightly out of alignment.

Figure 8 shows the result of the aggregation: The aggregation of the tokens have alternately lower and higher transparency. The parts of lower transparency show tokens

derived from a few simulations runs that are out of alignment. Large parts of tokens aggregate to higher transparency so that the observer can identify time lags by this kind of pattern.

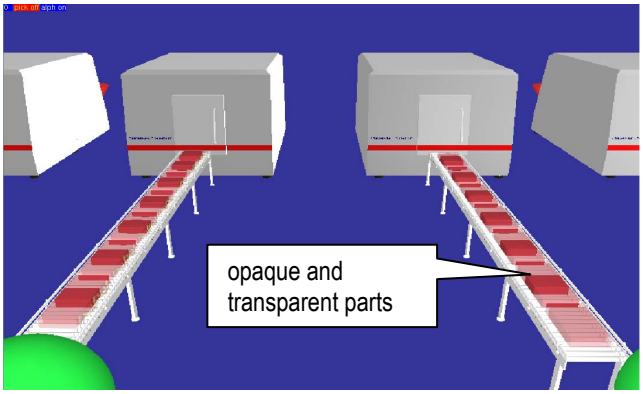

Figure 8: The conveyor belt of each simulation run is full with tokens. Slight displacements of the tokens results in alternating parts of lower and higher transparency.

4.2.4 Observation of Multiple Entities

The rendering technique aggregates several simulation runs and shows the result in a single 3D scene. The visual aggregated tokens make a concurrent observation of different conveyor belts possible. The observer compares how the tokens pile up on different conveyor belts. The aggregation is a good mean to observe stochastic influence and analyze multiple entities at the same time.

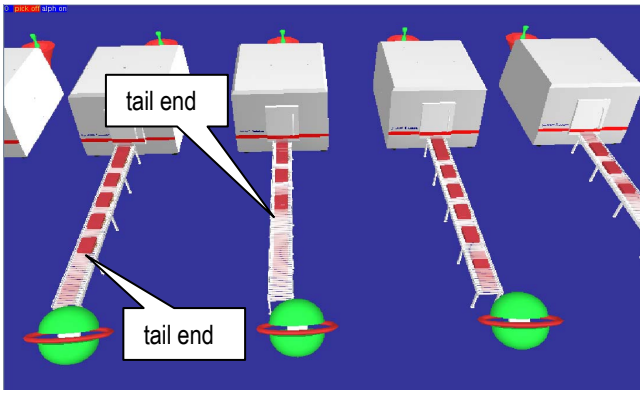

Figure 9: Observing the tokens of four conveyor belts. Each conveyor belt is an aggregation of 16 simulation runs.

An observer of the scene in Figure 9 analyses the four queues. He observes that the tailback of the middle conveyor belt is shorter than the tailbacks of the neighboring conveyor belts. Furthermore he observes that the tokens after the tail end of the left conveyor belt are less transparent than those on the right conveyor belt. This means that only a few simulation runs have longer tailbacks.

4.3 Aggregation of Tokens with Different Size

The preceding section shows tokens with uniform size and color. The transparent tokens of a single simulation run aggregate to tokens of the same shape. Now tokens are investigated which have the same color but different sizes. We use a small and big box for the tokens that are transported across the conveyor belt.

Figure 10 shows the aggregation of small and big tokens of all simulation runs. The front of the conveyor belt shows aggregated small tokens followed by the aggregated big tokens. Because both tokens have hardly any transparency, it can be concluded that all simulation runs have tokens there.

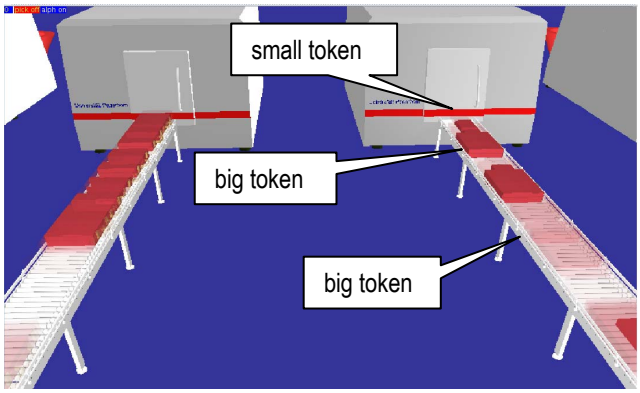

Figure 10: In front of the conveyor belt the aggregation of small and big tokens derived from all simulations runs. The end of the conveyor belt shows the aggregation of big tokens derived from few simulation runs.

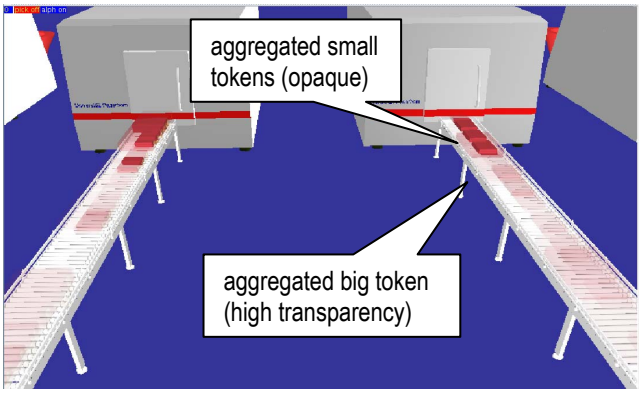

Figure 11: The front of the conveyor belt shows aggregated small tokens derived from many simulation runs and a big token derived from a few simulation runs.

Figure 11 shows aggregated small tokens rendered with low transparency and a big token with high transparency in front of the conveyor belt. The reason is that most simulation runs have a small token there and only a few simulation runs have a big token located there. The example shows that tokens of different size can be aggregated. We can distinguish between tokens of different sizes.

4.4 Aggregation of Colored Tokens

The preceding section shows token of different size but a single color. This section investigates tokens with two colors (blue and yellow). Figure 12 shows in front of the conveyor belt the aggregation of big yellow tokens and at the end of the conveyor belt the aggregation of blue tokens.

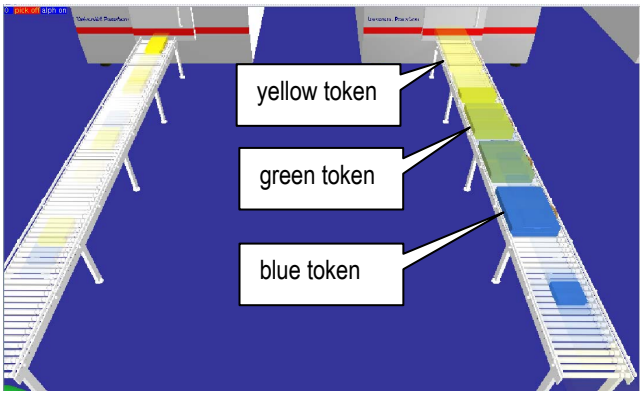

Figure 12: The conveyor belt transports only yellow and blue tokens. The green token in the middle is an aggregation of yellow and blue tokens.

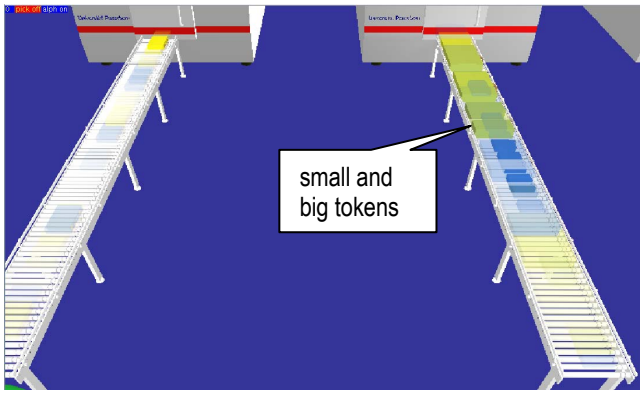

Figure 13: Aggregation of colored token of different size.

Green tokens are located in the middle of the conveyor belt. The simulation model does not include green tokens. The green color is derived from the aggregation of transparent yellow and blue tokens. Some simulation runs transport yellow tokens and some transport green tokens. It is unclear if this effect is beneficial. It can be confusing if the user has to think about the aggregation of colors. The situation gets more complex if the number of colors increase. Typically the user does not know the aggregated colors and therefore he cannot deduce the composition of the simulation runs. Therefore we think the usage of colors is reasonable only for a few numbers of colors.

Figure 13 shows aggregated small and big tokens at the same location. The small tokens shine through the big tokens.

5 IMPLEMENTATION

For the parallel simulation and rendering a Linux based PC-cluster is used and for controlling an external visualization-client is used. The simulation-kernel and the visualization-client are written in Java. The rendering-system is implemented in C++, using MPI and OpenGL. The same C++-program is started on all cluster nodes, but every node is specialized by its rank. The rank of a node is similar to an ID. On the PC-cluster there are three types of nodes: a master node, static rendering nodes and dynamic rendering nodes. The external client is connected with the master node. The master node is unique and sends the incoming commands from the client to the static rendering nodes, composes the pictures from the static and the dynamic rendering nodes and sends the composed pictures to the client.

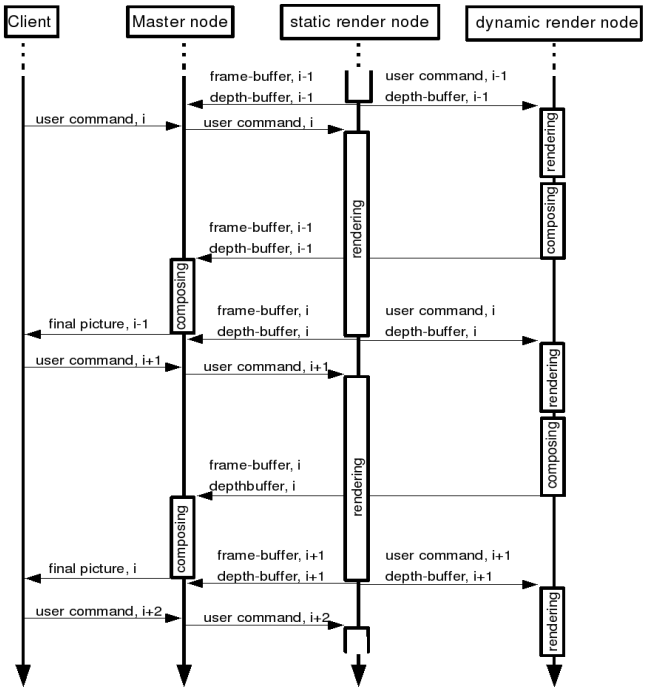

Figure 14: Communication scheme

It is assumed that the number and the complexity of static objects is much larger than the dynamic objects and that the static objects are constant for all simulations. After receiving a user-command the static rendering nodes renders the static parts of a simulation, for every frame. The frame- and the depth-buffer are sent to the master node. Additionally the depth-buffer and the last user-command are sent to the dynamic rendering nodes. The nodes wait for new commands, after finalizing sending.

Every dynamic rendering node is connected to a simulation. They receive the signals of the simulation-kernel, the depth-buffers and the user-commands from the static rendering nodes. After composing the depth-buffers the dynamic rendering nodes evaluates the incoming command and compute a picture of the dynamic objects of the simulation. After finalizing the rendering process, the depth-buffer of the rendered frame is compared with the composed depth-buffer of the static rendering nodes. In this way, hidden parts will be erased. The pictures of the dynamic rending nodes are composed hierarchically. Initially the alpha-values of all pixels will be changed, in dependency of the number of shown simulations. The alpha-value fixes the transparency of a pixel – zero corresponds to clear and one corresponds to opaque. If n simulations are shown, the initial alpha-value is set to 1/n. If two objects are placed at the same position, the alpha-values will be added and the color-values will be mixed. The depth-values will not be changed. The variation between different simulations can be visualized, in this way. The final picture, composed in this way, will be sent to the master node. Here the static and the dynamic pictures will be merged and sent to the client.

6 CONCLUSION AND OUTLOOK

The paper presents a 3D rendering technique to aggregate multiple simulation runs in single 3D scene. The technique allows the analysis of an interactive 3D environment with its full stochastic width. It allows direct judgment, how typical the observed visualization of the simulated simulation model is. Further research is necessary to develop techniques for using tokens of different sizes and color.

REFERENCES

Akpan, J. I., and R. J. Brooks. 2005a. Practitioners' perception of the impacts of virtual reality on discrete-event simulation, In *Proceedings 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 *Proceedings 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 *Proceedings 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 *Proceedings 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. D³FACT insight – An immersive material flow simulator with multi-user support. In *Proceedings of the 2005 Summer Computer Simulation Conference*, 239-242

Fischer, M., B. Mueck, K. Mahajan, M. Kortenjan, 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 *Proceedings of the 2005 Winter Simulation Conference*, 1921-1930.

Fischer, M., C. Laroque, D. Huber, J. Krokowski, B. Mueck, M. Kortenjan, M. Aufenanger, W. Dangelmaier. 2007. Interactive Refinement of a Material Flow Simulation Model by Comparing Multiple Simulation Runs in one 3D Environment. In *European Simulation and Modelling Conference* (ESM 2007), 499-505. EUROSIS.

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 *Proceedings of the 2000 Winter Simulation-Conference,* 1676–1683.

Hybinette, M. 2004. Just-in-time cloning. In *Proceedings 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 *Proceedings of the 2006 Winter Simulation Conference*, 84-95.

Kamat, V. R., and J. C. Martinez. 2000. 3D Visualization of simulated construction operations, In *Proceedings of the 2000 Winter Simulation Conference*, 1933-1937.

Law, A. M., and W. D. Kelton. 2000. *Simulation Modeling and Analysis*, 3rd Edition. McGraw-Hill Higher Education.

Schulze, T., S. Straßburger, and U. Klein. 1999. Online data processing in simulation models: New approaches

and possibilities through HLA. In *Proceedings 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 *Proceedings of the 2005 Winter Simulation Conference*, 1949-1957.

VDI-Richtlinie 4499: Digitale Fabrik – Grundlagen. VDI Verlag Düsseldorf, Düsseldorf. 2006.

AUTHOR BIOGRAPHIES

WILHELM DANGELMAIER studied Mechanical Engineering at the University of Stuttgart, Germany. In 1981, he became director and head of the Department for Corporate Planning and Control at the Fraunhofer Institute for Manufacturing. In 1991, Dr. Dangelmaier became Professor for Business Computing at the Heinz Nixdorf Institute. In 1996, he founded the Fraunhofer Center for Applied Logistics (Fraunhofer ALB). His e-mail address is <dangelmaier@hni.upb.de> and his Web address is <http://wwwhni.upb.de/cim>.

MATTHIAS FISCHER studied computer science at the University of Paderborn, Germany. Since 1995 he has been a research assistant at the Heinz Nixdorf Institute. In 2005, he received a Ph.D. for his work on distributed virtual environments. His research interests are computer graphics, real-time rendering algorithms, and distributed computing. His e-mail address is <mafi@upb.de> and his Web address is <http://www.upb.de/cs/mafi>.

DANIEL HUBER studied industrial engineering at the University of Paderborn, Germany. Since 2005 he is a research assistant in the group of Prof. Dangelmaier, at the Heinz Nixdorf Institute of the University of Paderborn. His main research interests are material flow simulation, simulation visualization and complexity reduction of simulation models. His e-mail address is <huber@hni.upb.de> and his Web address is <http://wwwhni.upb.de/cim>.

CHRISTOPH LAROQUE studied business computing at the University of Paderborn, Germany. Since 2003 he has been a Ph.D. student at the graduate school of dynamic intelligent systems and, in 2007, received his Ph.D for his work on multi-user simulation. Since then he is team leader of the “simulation & digital factory” at the chair of Business Computing, esp. CIM. He is mainly interested in material flow simulation models and the “digital factory”. His e-mail address is <laro@hni.upb.de> and his Web address is <http://wwwhni.upb.de/cim>.

TIM SÜß studied computer science at the University of Paderborn, Germany. Since 2007 he is a Ph.D. student in the group of Prof. Meyer auf der Heide at the Heinz Nixdorf Institute. He works for the project AVIPASIA which main topic is the visualization of parallel simulations. His main research interests are the development of parallel rendering algorithms. His e-mail address is <tsuess@upb.de> and his Web address is <http://wwwhni.upb.de/alg/mitarbeiter/tsuess>.