Proceedings of the 2004 Winter Simulation Conference
R .G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds.

USING DYNAMIC MULTIRESOLUTION MODELLING TO ANALYZE LARGE MATERIAL FLOW SYSTEMS

Wilhelm Dangelmaier
Bengt Mueck

Heinz Nixdorf Institute
University of Paderborn
Fürstenallee 11
33102 Paderborn, GERMANY

ABSTRACT

The interactive, simulation-aided analysis of material flow systems is often done with the help of virtual reality. If a user wants to influence the simulation run, the simulation and the visualization have to be computed simultaneously. Large, detailed simulation models, which can not be calculated fast enough, can not be analyzed interactively. This paper presents a method which only computes those parts of simulation models that are visited by the user. If a user shifts its attention, the area being simulated in detail is updated. Since the major part of the simulation is not detailed, the necessary calculating complexity is reduced. On the basis of models allowing a multiresolution simulation methods are introduced which regulate the level of detail based on the evaluation of the user's attention. Changing the levels of details leads to the exchange of models with different level of details.

1 INTRODUCTION

The simulation of production processes is a common tool in order to ensure strategic as well as tactical decisions. If a model is once developed for a process, different versions of this process can be easily analyzed. Questions like "How much more can I produce when using one more fork lifts?" can be easily answered. Production programs can be checked and verified with simulations. Can all customers be supplied on time? Are all preliminary products available on time?

People are surrounded by a three-dimensional world. They can intuitively understand this world. The objectives of planning processes, e.g. production facilities, are also three-dimensional objects. Virtual, three-dimensional, computer-generated worlds have become the state of art (cp. Dangelmaier and Mueck 2003). These three-dimensional environments are more and more introduced into the analysis of material flow simulations. The methods serve the user for a higher immersion and therefore lead to a better understanding of the simulation process. Mistakes can be detected faster. Incorrect planning can be avoided. The examination of layout can be integrated. The results can be communicated more easily to people who are not simulation experts.

For this purpose simulation models have to be developed first. These models are executed in a simulator and they are analyzed either during the run-time of the simulator or after the simulation. An analysis during the run-time offers the possibility to influence the simulation interactively. The observer is therefore integrated in the simulation. This helps him to understand the process better. So he can rapidly detect and improve problematic interrelations or processes.

The rate of the simulation execution depends on the number of the arithmetic operations being necessary for the computation of the simulation. If the number of arithmetic operations per time unit (of the simulation time) exceeds the number of operations which the computer can process in the real time in a larger interval, the computation of the simulation takes longer than the computed simulation time. A simulation computation being too slow eludes an interactive analysis and therefore its advantages.

The number of the required arithmetic operations per simulation time unit depends on the size and complexity of the system. A high level of detail in a complex model is therefore contrary to the fast computation of a simulation. The highly-detailed execution of large, detailed simulations is not feasible when the simulation time is faster than the calculation time.

If the simulation is to be analyzed by a user in a virtual environment, a synchronous progress of the simulation time with the user's clock is desirable for a good immersion of the user. If the simulation is computed too fast, the computer can have waiting periods and therefore decelerate artificially the computation. For the case that the simulation is not computed fast enough, this paper presents a method for the acceleration of the computation.

The starting point of this work was the case, that the user of the system was not able to observe the whole scenery with a consistent attention in an interactive, virtual environment. Areas being far away from the user or being occluded by other objects can not be seen very well. The idea of the method was to simulate only those areas with a high level of detail which can be seen very well. The areas which can not be seen very well are to be simulated with a low level of detail and therefore with less calculating complexity.

In the method of this paper those areas follow the user when executing the simulation. By following the user‘s movements and view the illusion of a continuously and highly-detailed simulation is created. This can be done without the computing power required for a continuously and highly-detailed simulation.

In return a low-detailed simulation of areas which the user does not visit at that time is accepted. Since the user is not able to see these areas, he does not realize that the simulation is only computed with a high level of detail in his direct environment.

This leads to inexact results. However, these results are better than those of a complete non-detailed computation. If large, complex simulations, which can not be developed with common methods of an interactive analysis because of the high demand for computing time, are to be made available for the user, the inexact results represent a possible compromise.

The computing of the simulation is influenced by the user’s movements. Therefore, the simulation can have different results, if the user takes different ways through the virtual environment. For this purpose simulation models working with different levels of detail are required (see section 4). On the one hand methods which determine at which time an increase of detailing is executed (see section 5) and on the other hand methods which execute the switch between the different levels of detail (section 6) are required. The experiences being made with a prototype to execute such systems are presented in section 7.

2 SIMULATION OF MATERIALFLOW SYSTEMS

For the simulation of material flows token-based simulation systems, which function like Petri-Nets, are used. A manufacturing cell is reduced to the number of resources being necessary for manufacturing and to its processing times and malfunction periods and it is modeled as a transition (in the following being called static objects). A token represents a part or a resource being necessary for the production (e.g. human resource or a floor conveyor).

The state of the system is described by marking static objects and a number of future, time-marked events.

For the computation of these models the events being in the nearest future are taken successively from the event-memory and are processed on the basis of a calculation specification, which can shift marks and produce new events. An event can be initiated when e.g. a token enters a machine. This event then produces a new event for the point of time in the future, when the mark leaves the machine.

A number of commercial tools (e.g. Arena, Automod, Taylor ED or eM-Plant) are offered in order to execute such token-based simulations (cp. Klingstam and Gullander 1999). Despite the hierarchical approach for modeling large systems, the simulations are always calculated with the highest level of detailed in these systems. Calculation specifications for lower-detailed models, which e.g. combine some manufacturing cells, do not exist.

In recent years tools facilitating an analysis or even a modeling in a three-dimensional environment were integrated. The demand for and the benefit of three-dimensional visualizations for the analysis of simulation models are therefore approved from this side, too.

3 MULTIRESOLUTION MODELING

There are some examinations for modeling, maintaining and coupling models in different resolutions (e.g.. Davis and Bigelow 1998 or Reynolds and Natrajan 1997). In these approaches there are models of one object in different levels of detail, between which a switching is possible during the run-time of the simulation – if needed (especially when two partial models want to communicate on different levels of detail). If a lot of switches are successively done, inconsistencies can occur. On the basis of this problem Natrajan (Natrajan 2000) proposes to work with only one description of the state and to provide interfaces on each level of detail for interactions with partial models, which are available on different levels of detail. The information being necessary in the interfaces are only generated by aggregation or disaggregation when needed. Since only the interfaces information are transformed, the description of the state is not affected by these transformations. Thus, a sequence of aggregations and disaggregations does not change the state of an object.

An alternative approach is presented by Kim et al. with the system Entity Structuring and Model Base Management Approach (SES/MB) (cp. Kim et al. 1992). They collect partial models with different levels of detail in a model base for the working with different levels of detail. The SES/MB contains a tree comprising possible compositions of partial models for an overall model for the composition of simulation models, which are based on the models of the model base. Models with different levels of detail can be generated from this tree and the MB. The level of detail is determined before the simulation is computed. A level of detail which changes dynamically to the run-time is not planned.

These approaches do not use the user’s view as stimulation for changing the resolution of individual objects during the execution time. Their objective was not to reduce calculation time.

Systems changing dynamically the resolution during the simulation stimulated by the user’s view are only marginally known for special applications (e.g. Carlson and Hodgins 1997). The results are hardly applicable to the general modeling, dynamic and resolution variable simulation of discrete models.

4 MULTIRESOLUTION MODELING OF MATERIALFLOW SYSTEMS

Token-based material flow models (see section 2) are to be used as the basis. These models have to be completed by additional information for the computation on different levels of detail.

4.1 More Detailed Models

If an area of the simulation is to be simulated on different levels of detail as intended in the vision, different simulation models have to be used. In this approach the models are determined by the modeler before the simulation is run. In this way a static object is described by a number of linked, static objects. An object can describe a whole manufacturing line on a low-detailed level and each manufacturing cell is described by a static object on a more detailed level. On the other hand those objects can contain models for an even more detailed simulation. The result is a hierarchy regarding the used models. In contrary to a common hierarchical modeling these objects, which only capsulate the more detailed models, are autonomously operational on each level. During the run-time different models can be dynamically connected to an overall model.

4.2 Connections of Mark Flows across Several Levels

The static objects pass on marks along the interaction edges during the simulation run. If an object receives a mark being simulated more detailed at this moment, the mark has to be passed on to the more detailed model. For this purpose a method has to be provided by the multiresolution model. Vice versa objects of more detailed levels have to be able to send marks to static objects from the next less detailed level. For the modeling we take a top-down-approach as the basis.

The less detailed models are modeled before the detailed models. They can exist autonomously. We dissociate us from the idea of autonomy for the more detailed model when modeling the mark flow from the detailed to the less detailed level, and we allow the modeler to set references to the objects of the next higher level for the mark flow.

For one step a communication is not necessary. The marks can be passed on step by step up or down. If two static objects of one level are simulated with more detailed models at one point of time, the communication of the more detailed models takes place on the less detailed level with the method described above. There is no need for additional methods of communication between the more detailed models (see figure 1).

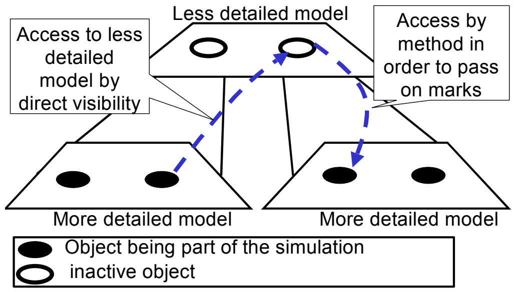

Figure 1: Communication between Two More Detailed Models

4.3 Information for Methods for the Generation of State

The state of a model is described by setting marks on static objects and by the state of the event queue. If it is to be switched to a more detailed model for an area which has been simulated by a static object so far, an initial state has to be generated for the more detailed object. Vice versa the state of the object has to be generated when switching from a detailed model to a static object. For this purpose various methods are discussed in section 6. Since some of these methods require additional information in the simulation model, those are listed here.

Therefore all information being required additionally in a model for a resolution variable simulation are collected. Figure 2 presents an overview of the cooperation of the methods.

4.4 Additional Information

The additional information described in section 4.1-4.3 for the work with resolution variable models require more extensive models than common simulations require. In order to be able to estimate the additional complexity, in addition to a resolution variable model a non-resolution variable model is to be used to draw a comparison. For a comparison a model corresponding to the resolution variable model on the most detailed level is to be used. Models which do not have this resolution describe the facts with less details and therefore it is difficult to compare them with the resolution variable model.

Let N_D be the number of static objects on the most detailed level. If it is still assumed that at least 2 static objects describe one model for a less detailed static object from one level of detail to the next one, the height of the hierarchy H_H is logarithmically limited by N_D:

$$H_H \leq \log_2(N_D)$$

More than half of all static objects N_A are therefore on the lowest level:

$$N_A < \frac{1}{2} N_D$$

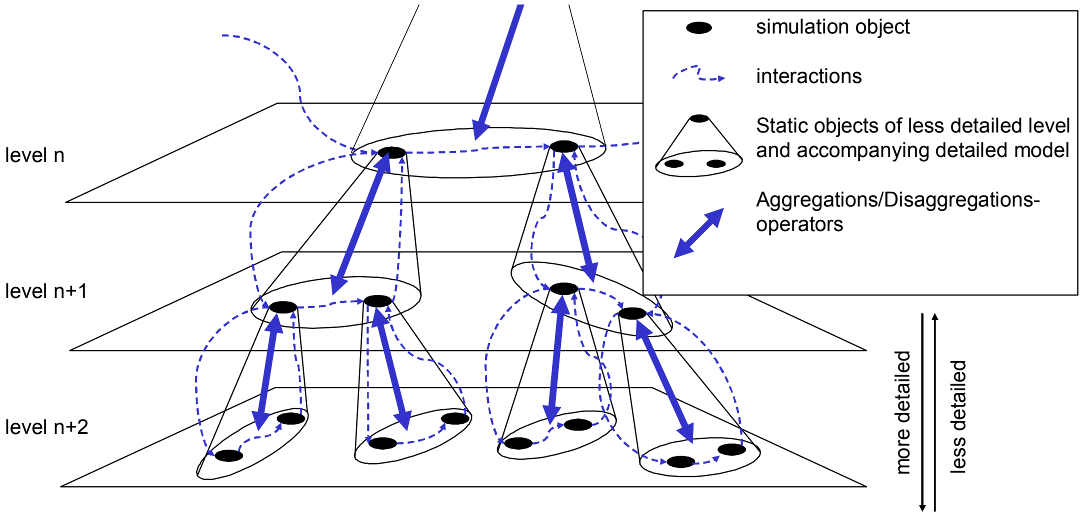

Figure 2: A Multilevel Model

Since exactly one detailing method (see section 4.2) and one method to pass on marks to more detailed models (see section 4.3) belong to each object which is not on the most detailed level, the number of methods for the state generation N_G and the number of methods to pass on marks N_M can be estimated with the number of objects which are not on the most detailed level:

$$NG \leq N_A < \frac{1}{2} N_D$$

$$N_M \leq N_A < \frac{1}{2} N_D$$

As a result the additional complexity which is produced by resolution variable simulation models is limited in a linear way by the number of objects of the comparison model.

5 STIMULATING DIFFERENT LEVELS

The level of detail of the object simulation is to change during the run-time of the simulation in a dvdS. The different levels of detail of an object are hierarchically given by various simulation models in a dvdS. A number of models being used in the simulation at this moment are to be chosen. These models have to be activated and deactivated.

Two questions arise: How often does the resolution have to be adjusted and where does an aggregation or a disaggregation have to occur. Section 5.1 deals with the first question and section 5.2 with the second one.

5.1 Quantity of Calculations

At one point in time the user is at a certain place. The parts of the simulation around him are to be simulated in detail. If the user wants to move, the user always has to stay in the highly detailed area when moving or after the movement for the continuous impression of a highly-detailed simulation. The size of the less detailed area is to be as small as possible. The more detailed areas are to be as large as possible. This demand is contrary to the required calculation time. The areas around the user can be adjusted before or after the changing of his position. In order to guarantee this, a conservative as well as an optimistic procedure are appropriate. If a user wants to change his position, the procedure adjusts the level of detail first and then changes the position. The optimistic procedure allows the user to move freely and regulates periodically the activations. Because of the reduced number of tests the highly-detailed area has to be larger.

5.2 Indications for Changing the Resolution

If the resolution of a model is to be adjusted to the user at one point in time, each static object being part of the simulation at this time has to be checked if it has the appropriate resolution. This is done with an evaluation of the object on the basis of the user's attention which can be determined by the following indicators:

5.2.1 Indication by Distance

If a user is far away from an object, the object seems to be smaller. Therefore, it does not have to be simulated with a high level of detail.

5.2.2 Indication by Occlusion

If an object can not be seen since it is behind other objects, the evaluation can be lower with this indicator. This indicator is appropriate, if the material flow stretches across several rooms/buildings. If the objects stand in a detached way in the room, just a little movement of the user can result in large variations of the evaluation (visible / not visible).

This can lead to increasing aggregations and disaggregations which are not desirable.

5.2.3 Indication by Angle of View

If something happens behind the user, he is not able to see that. The evaluation of those objects can be reduced. If the user can turn fast, this indicator can change quickly.

5.2.4 Indication by Link

The static objects are linked together by edges in the material flow systems which are to be examined. Linked objects have an influence on each other in the simulation run. If one object is marked by a high attention of the user, the user shows a lot of interest in this object. This object is therefore to be simulated in detail. The quality of the simulation also depends on the input data. That's why the evaluation of objects which have a logistical connection with objects having a high evaluation regarding the other indicators is improved.

5.2.5 Bringing the Indications Together

The indicators are joined to an evaluation of the static objects. If a static object exceeds the limit of a disaggregation, a disaggregation is stimulated. On the contrary, meaning that the indicators point to an aggregation, the case seems to be more difficult. In the hierarchical modeling single objects can not be aggregated to less detailed ones. Only whole models can be aggregated. In order to stimulate this, the evaluations of all objects which are on this level of the model have to be evaluated. If an object is simulated by a more detailed model at this point in time, either it has to be waited until the model is aggregated or an aggregation at the has to be stimulated.

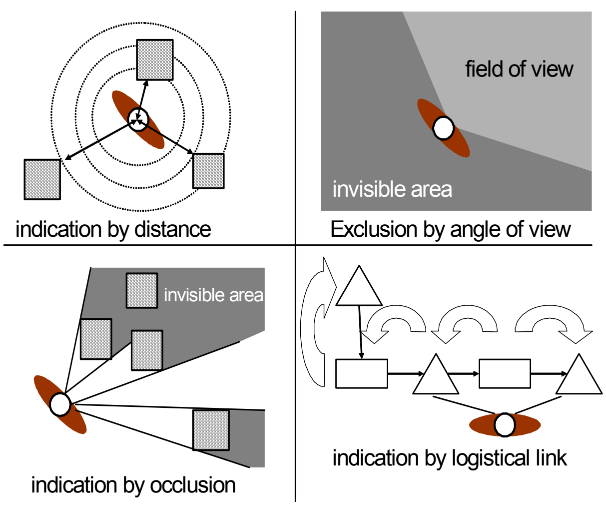

Figure 3: Various Indicators for the Determination of the User's Attention to an Object

6 GENERATING DATA

If an object is to be disaggregated or a model is to be aggregated, a state has to be generated for the newly activated description. For this purpose, methods have been developed:

6.1 Re-Simulation

If a model is active, a state is described by its occupancy and the event queue. If now a more detailed model is activated for the position, the model of this stimulated level is simulated with the saved events from the time of the last activation until now. Vice versa when having a decreasing resolution, the events which have occurred for the position to be activated since the last activation are to be calculated again.

The alternative descriptions do not have necessarily the same results. The events to be stimulated do not have to be those of the other description. This procedure can imply inconsistencies which can be identified and partially corrected.

6.2 Time-Limited Re-Simulation

If the last activation was done a long time ago, a lot of events have to be simulated afterwards. With this method only a part of the past which is sufficient for a leveling out of the model is simulated in this case.

The quality of the generated states depends on the duration of the simulated period. If the period is too short, the model is not correctly initiated. Let t_a be the starting time of simulation. If t_a corresponds to the last activation time of the object to be activated or of the partial model to be activated, this method is not equivalent to the re-simulation.

A further extension of the simulation time in the past beyond the time of the last activation will not improve the quality of the generated state. If a shorter re-simulation time is chosen, the generated solution will be of lower quality. It is possible to choose between a high quality and a fast calculation with the chosen time for the re-simulation. Therefore, the method is parameterizable regarding quality and required calculation time for the duration of the period to be simulated.

An example: A model is to describe a manufacturing line through which parts sequentially go. The line consists of several machines being described by their cycle time and their reliability. A cache memory is to be between the machines. Incoming parts are processed as external inputs for this line. Now this model is to be activated and the last activation should have been done a long time ago. All parts having gone through the manufacturing line since the line had been active for the first time have to be simulated when executing a complete re-simulation. During a time-limited re-simulation only a part of the period is simulated. If this period is long enough, so that the states of the machines and the stock have leveled out, the state for the simulation is almost as good as the state of the whole re-simulation. However, the required calculation time is shorter.

With this method the potential inconsistencies are equivalent to those of the complete re-simulation. In order to solve them the same methods are appropriate.

6.3 Re-Simulation with Observation of Interactions

The re-simulation and the time limited re-simulation are methods for the generation of states which only calculate approximately consistent states. Even a correction can not make the states consistent in every case. If these errors are not acceptable in an application, those methods are not applicable. The errors being made by both of the methods mentioned above are due to the missing consideration of interactions. The models/objects to be activated are calculated without taking the environment into account. Regarding the re-simulation with observation of interactions the positions to be activated are not calculated without the rest of the simulation. If there are interactions from the model/object to be activated with another position, the model/object is integrated into the calculation of the position. If the position interacts with another position in a different way, this position has to be integrated from this time on, too. Therefore, the number of elements to be simulated increases strictly.

6.4 Direct Calculation

The modeler knows both the detailed model as well as the less detailed object. He also knows what the static objects of the more detailed model represent. Therefore, he has information about the model which is not included in the model. If the modeler can implement a method which initializes a more detailed model from a less detailed position or a less detailed position from a more detailed model in an analytic way, this knowledge should be used.

An example: A manufacturing line is simulated with several positions as a position on a level and as a sequence of single machines and intermediate storage on a lower level of detail. If now this construct does not have a detailed state and it receives a resolution message, the method distributing the parts being in the line among the machines is activated. If, apart from the parts in the line, the last processing times are also known, they can be used for the distribution among the detailed machines. Vice versa in the case of a decreasing resolution event, the parts are transferred from the machines to the lines and they are marked with the past processing times corresponding to the position of the machine in the line where the parts had been (cp. Figure 4).

6.5 Evaluation of the Methods

The presented methods have specific advantages and disadvantages which differ in required calculation time, quality and generated states.

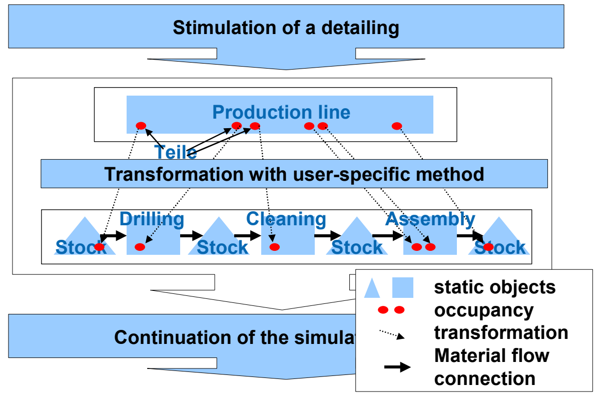

Figure 4: Direct Calculation of the State

6.5.1 Re-Simulation

The re-simulation can be automatically supported by the simulator. There is no need for additional effort during the modeling. The calculation complexity can increase extremely since the last activation could have been a long time ago. The quality of the calculated states is likely to be good.

6.5.2 Time Limited Re-Simulation

In the case of a time limited re-simulation the calculation complexity is lower than in case of a complete re-simulation. Especially the worst case of an re-simulation – a complete simulation of all levels for all points in time – can be avoided here. The quality for long, simulated periods equals to the quality of a complete re-simulation. In case of short, simulated periods the quality can be managed with the simulated time. The modeling complexity is low.

6.5.3 Re-Simulation with Observations of Interactions

The advantage of the re-simulation with observations of interactions is the high quality of the generated states. A major drawback of the method is the high calculation complexity. It is possible that a lot of parts of the model have to be simulated due to the sequential integration of an increasing number of positions. Especially if the last activation was a long time ago, a lot of positions and therefore events have to be potentially simulated. A gain in calculation time as intended in the outline is not possible in this way.

6.5.4 Direct Calculation

In case of a direct calculation the modeler has to specify a method for the state transitions. Therefore, the modeling complexity is higher than other methods require. The accuracy and the calculation complexity depend directly on this method. Since the knowledge of the modeler is integrated by the simulation model, the results are likely to be good (if the modeler is experienced).

7 IMPLEMENTATION

For the validation of the methods a part of the proposed methods was implemented. A formal description was developed for the modeling of multiresolution models containing all additional, required information (see section 4). Based on this description a XML-based language being was used for the development and processing of the models. A parser preparing the required data structures for the simulator was implemented and integrated into the overall system. Therefore, the user has to encode the model with the language and can pass it on to the simulator.

The stimulation of switching processes is based on the distance of the user to the respective object and the angle of view in this prototype (see section 5.2.1). The method re-simulation is used for the generation of detailed states in the current implementation (see section 6.1). Further methods are in the implementation.

An example: For the presentation of the implementation an exemplary model was chosen. Parts have to go from the goods receipt (WE) through a warehouse (L), through a processing (B) to the goods issue (WA). Both the warehouse W as well as the processing have more detailed models (see figure 5).

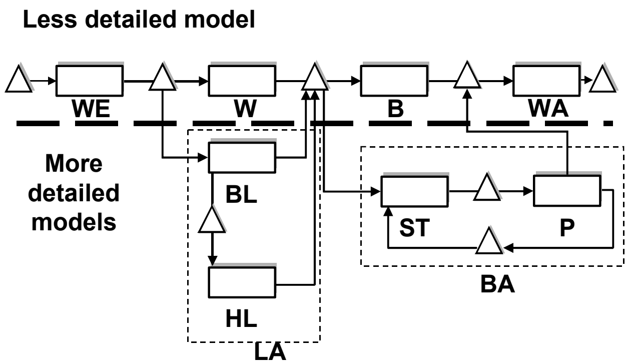

Figure 5: Schematic Overview of the Demonstration Scenario

The static objects are visualized with individual 3D-Models in the implementation. The occupancy of the static objects is visualized with little cubes. The screenshot in figure 6 shows a user who keeps enough distance to see all 4 static objects on a low detailed level.

If the user comes up to these objects, detailings are initiated in case of a sufficient approach. Figure 7 shows the state after an disaggregation of the warehouse. A new more detailed model of the warehouse was activated and the state of the model was generated.

8 CONCLUSION

Based on the observation, that in complex models the user can only pay attention to a part of the model, in this work methods computing only the simulation of those parts in

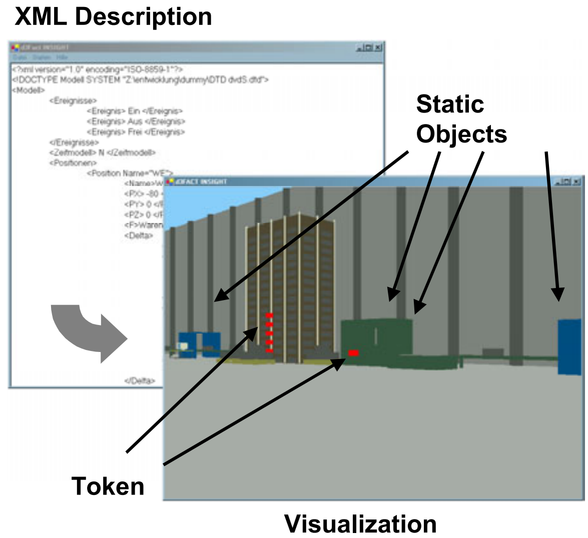

Figure 6: The User is Far Away

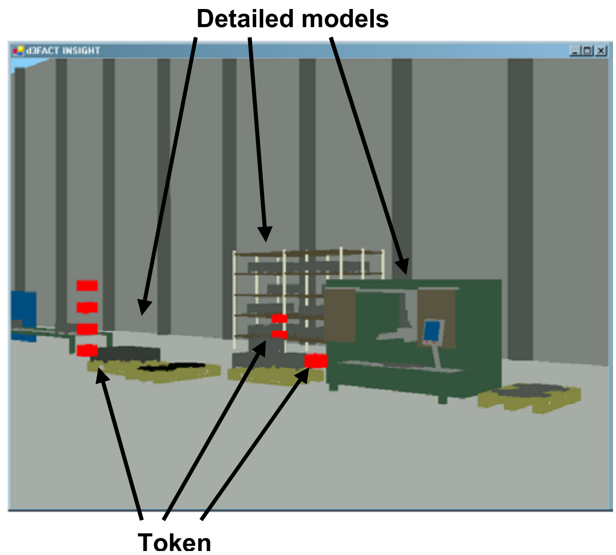

Figure 7: After an Disaggregation Prozess one Object is Simulated with 2 Objetcs.

detail which can be seen by the user were developed. Areas which are not really observed by the user are simulated with a low resolution and with less calculation complexity.

More complex models can be interactively analyzed due to the tolerance of inaccuracies. If the user changes his position or angle of view, the level of detail is updated according to the changed attention to the objects. Despite the reduced calculation complexity the user still has the impression of an environment being simulated constantly with high detail.

In order to guarantee this, there are various detailed models (or hierarchies of models) for the objects of the simulation. A part of the models has to be activated at one point in time. Those activated models form the simulation model to be computed. One focal point was the determination of the activated models. This is done by a number of indicators. The analysis comprises the distance of the user to an object, potential occlusions of the object which can be caused by objects between them and the user, the limitation of the user's angle of view – the user is not able to see objects behind himself – and logistical interrelations in the simulation – if a machine is interesting for the user, he is also potentially interested in previous and following machines.

If the user's attention increases or decreases, a switch between the models is stimulated with the help of indicators.

The sub-models to be activated now have not been used during the simulation run or they have not been used for a long time. Their state is undefined. Before the integration into the simulation a state has to be generated which reflects the actual simulation state as precisely as possible. The state of the model to be activated or the list of the past influences on this can serve as the data base. For this purpose various methods were developed and evaluated.

The multiresolution simulation models are more complex than common models. Additional models with different levels of detail and information for activation/deactivation as well as methods for the generation of initial states have to be additionally included.

Within this work methods for pooling these additional methods to modules were determined for the efficient modeling and maintaining of multiresolution simulation models. Therefore, the complexity for modeling and maintaining being additionally produced by complex models could have been reduced. A prototype for the validation of the methods was implemented. The prototype proves the completeness of the methods as well as the potential of the reduction of calculation time.

REFERENCES

Carlson, D. A. and Hodgins, J. K. 1997. Simulation Levels of Detail for Real-time Animation. In *Proceedings of Graphics Interface 97 (GI'97)*, Canadian Human-Computer Communications Society

Dangelmaier, W. and Mueck, B. 2003. Simulation in Business Administration and Management - Simulation of production processes, In *Applied System Simulation: methodologies and applications*, ed. M. S. Obaidat and G. I. Papadimitriou, 381-396, Norwell, MA: Kluwer Academic Publishers

Davis, P. K. and Bigelow, J. H. 1998. *Experiments in Multiresolution Modeling (MRM).* RAND MR-1004-OSD, The RAND Corporation, Santa Monica, CA

Kim, T. G.; Lee, C.; Christensen, E. R. and Zeigler, B. P. 1992. System Entity Structuring and Model Base Management, In *Proceedings of Conference on Variable-Resolution Modeling*, ed. P. K. Davis and R. J. Hillestad, 96-101, RAND CF-103-DARPA, The RAND Corporation, Santa Monica, CA

Klingstam, P. und Gullander, P. 1999. Overview of simulation tools for computer aided production engineering, In *Computers in Industry*, 38: 173-186

Natrajan, A. 2000. *Consistency Maintenance in Concurrent Representations*, Phd. thesis, University of Virginia

Reynolds, P. F. Jr. and Natrajan, A. 1997. Consistency Maintance in Multiresolution Simulations. *ACM Transactions on Modeling and Computer Simulation*, 7: 368-392

AUTHOR BIOGRAPHIES

WILHELM DANGELMAIER studied Mechanical Engineering at the University of Stuttgart, Germany. Since 1981 he was director and head of the Department for Cooperate Planning and Control at the Fraunhofer Institute for Manufacturing (Fraunhofer IPA). In 1990 he became Professor for Facility Planning and Production Scheduling at the University of Stuttgart. In 1991, Dr. Dangelmaier has become Professor for Business informatics at the Heinz Nixdorf Institute; University of Paderborn, Germany. 1996, Prof. Dangelmaier founded the Fraunhofer Center for Applied Logistics (Fraunhofer ALB). His e-mail address is <whd@hni.uni-paderborn.de>. The web address of his working group at the Heinz Nixdorf Institute is <wwwhni.upb.de/cim/index_e.php3>.

BENGT MUECK studied computer science at the University of Paderborn, Germany. Since 1999 he is a research assistant at the group of Prof. Dangelmaier, at the Heinz Nixdorf Institute of the University of Paderborn. His main research interests are logistics systems and tools to simulate those systems. His e-mail address is <mueck@hni.uni-paderborn.de>.