d^3FACT INSIGHT: A SIMULATION-TOOL FOR MULTIRESOLUTION MATERIAL FLOW MODELS

Wilhelm Dangelmaier
Bengt Mueck
Christoph Laroque
Kiran Mahajan
Heinz Nixdorf Institute, University of Paderborn
Fürstenallee 11, 33102 Paderborn, Germany
{whd, mueck, laro, kiran}@hni.uni-paderborn.de

KEYWORDS
Multiresolution Simulation, Material Flow Simulation, Virtual Reality, Digital Factory

ABSTRACT

In a global economy, successful organizations constantly use innovative manufacturing methodologies to stay competitive in business. Among others, simulation is a tool which offers interesting perspectives from the manufacturing system optimization point of view. However, when a simulation model becomes large, and the entire model is simulated at a high level of detail or resolution, computing power tends to become a bottleneck. As a result, if the model is large it is seldom possible to calculate/simulate it in real time. Real time simulation is desirable if an interactive analysis of the manufacturing system is required within virtual environments. Secondly, in a virtual environment a user can view only a part of the simulation model. Hence, in our approach only this part is simulated in high resolution with high effort of computing power. The parts of the model, the user ignored or cannot view, are simulated on a rough level. Consequently, if the user moves, the area which is simulated in high resolution also moves with the user. This way he gets an impression of a simulation in high resolution. This also enables the user to analyze large simulations in real-time.

In this paper, we illustrate a method for detecting the user attention based on modeling approach. These detections will stimulate adjustments in the level of detail. After an adjustment the starting state of the newly activated models have to be generated. Methods to do this are shown. Most of these methods have been implemented in our simulation tool d^3FACT INSIGHT. Then, a short example of a multiresolution material flow simulation is shown followed by conclusions.

OBJECTIVES

Simulation of material flow systems is a well-known method to set up new production plants. It is easy to analyze different scenarios and to answer questions like: Will a faster machine raise my throughput? Besides this, processes and dependencies can be detected easily. For such simulations, virtual environments are often used to give the user a good impression of the model. He can move freely through the model and analyze production processes he is interested in. If he wants to modify or perform interactions with the model, the simulation needs to run simultaneously with the visualization. In this case it is not possible to base the calculation of the visualization on a trace file (Dangelmaier and Mueck 03).

The execution speed of a simulation depends on the size of the model, (which depends on the size of the system being modeled) the detail with which the model is being simulated and the speed of the computer calculating the model. Rougher models leads to a faster calculation. More detailed models need more calculation time. If the models are large and detailed enough, it is not possible to calculate them fast enough and further use them for analysis within interactive virtual environments.

Typically the user can only see a small part of a large scene. We simulate the area, which is surrounding him (and which he can view), with a high level of detail. The areas he is not viewing (or cannot view) are simulated on a low level of detail. If the user moves or turns around, the high detailed area follows the user. So the user gets an impression of a simulation which is calculated completely in high resolution. Because most of the simulation takes place in a rough level of detail, the required calculation time is reduced. As a result, bigger and more detailed models can be analyzed with this approach.

To implement this idea, first we need a representation of models which work in different levels of detail (at the moment these models have to be modeled individually by another modeler). During the simulation/execution one set of models is activated to represent the whole simulation-model. This activation of models has to be identified and if the user moves the activation has to change correspondingly. Besides this, indications of the required level of detail are also required.

For our material flow simulation the state of the system is preserved with tokens and their assignment to objects and an event queue. If the identification stimulates a change in the activation of different models, the state of

Proceedings 16th European Simulation Symposium
György Lipovszki, István Molnár © SCS Press, 2004
ISBN 1-56555-286-5(book) / ISBN 1-84233-106-x(CD)

the active model has to be generated. Methods will be described later in this article.

STATE OF THE ART

Some research has been done for modeling and simulating models in different resolutions (e.g. Davis et al. 1998 or Reynolds and Natrajan 97). 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). However, if a lot of switches are successively done, inconsistencies can occur. On the basis of this problem, Natrajan proposed 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 which is necessary in the interfaces is only generated by aggregation or disaggregation when needed. Since the interfaces information is only transformed, the description of the state is not affected by these transformations.

An alternative approach is presented by Kim, Lee, Christensen, and Zeigler with the System Entity Structuring and Model Base Management Approach (SES/MB) (cp. Kim, et al. 92). They collect partial models with different levels of detail in a model base for 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. In other words, in this approach, a level of detail which changes dynamically during 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. Secondly, their objective is not to reduce calculation time.

MODELING

To understand how we model multiresolution models we first take a short look on how we model traditional non-multiresolution models in our approach.

Non-multiresolution modeling

As mentioned earlier we are using token-event based systems to model material flow systems. So a model consists of positions which represent static resources like machines. The current state of the positions is described with the number of tokens currently assigned to the position (see Figure 1).

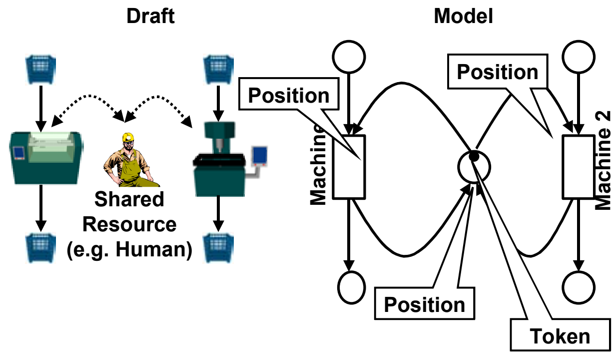

Figure 1: Modeling of production systems with a token based approach.

To change the assignments between tokens and positions we use events. An event occurs only at one specific position. It consists of a time and type. If the simulation time exceeds the time of an event, the event will be executed. A rule, which is part of the model, creates depending on the position, a type of event and the current assignment of tokens to the position a new assignment of tokens to the position and potential new events. So the connection in the material flow is modeled implicitly in the event interpretation rule.

E.g. the event which occurs, might be "Part leaves machine". The rule now might decrease the number of tokens in the position by one and create a new "Part enters the machine"-event at the following machine at the same time.

Multiresolution Modeling

To get an hierarchical multiresolution model each position can consist of a whole simulation model (see Figure 2). So if the position is simulated in more detail, the position will be turned off and the simulation model which is part of the position will be switched on. If the positions of the detailed model also contain more detailed models, this will ultimately lead to an hierarchical modeling approach.

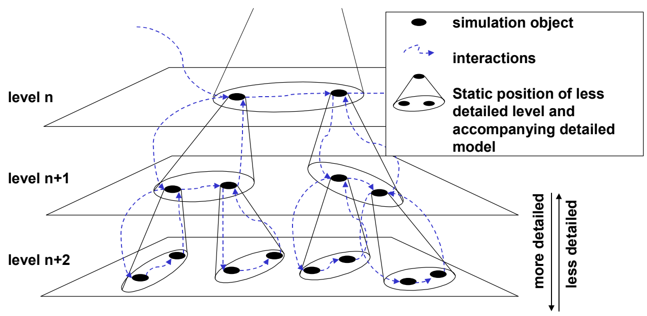

Figure 2: Multi resolution modeling

Events may occur on positions, which are currently not activated. The activation always guarantees that a more detailed model is activated. A transform rule is required to translate the events into one (or more) events for the more detailed model. If this model is also not activated the rule for this model will direct the event to a further more detailed level.

In our approach the detailed models can view the positions on a rougher level. Hence, they can directly generate events for these positions. A special translation rule is not required for this.

EXECUTION OF MULTIRESOLUTION MODELS

Activation of the right models

As mentioned earlier, one task is to activate the models which need to be simulated on a different level of detail. To do this, we developed 4 indicators as follows:

A. Indication by distance: The distance between the user and the object can be used as an indicator. Objects which are far away get low rating.
B. Indication by the direction of the view: The user can't see objects in his back. So objects with a great angle between the direction of view and the direct connection between the user and the model also get a low rating.
C. Indication by occlusion: Often the user cannot view the entire scene. Some objects may block the visibility of other objects. Objects the user cannot view see are rated low with this indication. To calculate this indication the 3D-representations of the virtual objects (which might consist of a lot of data) are required. If the user makes small movements the indication can change a lot. This could led to a lot of aggregation/dissagregation operations.
D. Indication by logistic dependencies: If the user takes a close look on one part of the simulation, it is logical that this part of the simulation is logistically connected with other parts of the model. With this indication, the rating of parts which are connected to parts which already have a good rating from other indications, will be increased.

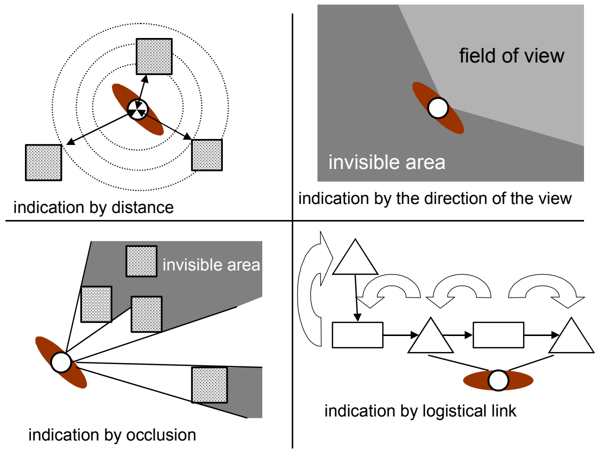

Figure 3: Different indications for the user attention

After calculating the indication of each active position, we decide whether a position has to be simulated in more detail (if a model is available) or a model has to be aggregated (if possible).

Generating states

When an aggregation or disaggregation operation takes place, a model or a position will be added to the global model. Vice versa a position or a model will be switched off. But the state of the newly added part represents the state when the part was active last (if it was used before). This state might not be the right one for the current simulation time.

A state for the newly activated position or model has to be generated. The first approach is to simulate again all events, which occur on the position/model since the last activation.

If the last activation was done a long time ago, a lot of events have to be simulated afterwards. With the second approach only a part of the past which is sufficient for leveling out of the model will be recalculated. The considered events are selected by the time difference between the actual simulation time and the time assigned to the event. So old events will not be considered. This leads to incorrect states but it can reduce the needed calculation effort enormously.

The re-simulation and the time limited re-simulation are methods for the generation of states which only calculate approximately consistent states. This is due to the fact that potential interaction with other parts of the simulation are not taken into account. If these errors are not acceptable in an application, those methods are not applicable. 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.

Instead of doing a re-simulation it is also possible to model specific functions, which translates the current state of the position/model, which will be deactivated, into a state for the active position/model. This method requires additional modeling efforts.

ESTIMATION OF THE REDUCTION OF NEEDED CALCULATION EFFORT

The needed calculation time to execute a model depends on the size of the model. If the user builds a hierarchical multiresolution model where at least each position of a rougher layer consists out of 2 positions of the more detailed layer, the size of each layer will be the half of the size of the more detailed layer. If the model has got l layers and p positions on the most detailed layer, the number of positions of the roughest layer will be less than $p/2^l$.

At one point in time only a small part of the simulation should be activated on the most detailed layer. Most of the Simulation be carried out on the roughest layer. Lets assume that the size of the detailed activated small model is s_d. Then the size of the whole activated model will be less than $s_d + p/2^l$. Remember that s_d should be small. So the activated model is something around $p/2^l$. Compared to the model the detailed layer, which has p positions, the size is decreased by 2^l . If we assume that the needed calculation time depends linear on the size of the model, the execution speed-up will be at least 2^l .

Up to now the calculation didn't take the needed efforts for the calculation of the activation and the generation of states into account. In the following we are showing a rough estimation of the additional needed calculation efforts to give the reader an idea about it.

Only the indication of the activated positions have to be checked. As mentioned before the number of activated positions is much less than the number of the positions on the detailed layer. So the needed calculation efforts are small compared to the efforts of calculating the whole model. The need time to generate the state of a new activated position/model depends on the method, which is used to calculate the state. If the calculation is carried out by methods, which are in the model, the calculation time will depend on these methods. The calculation is a local operation so the calculation time will not depend on the size of the overall model size. If the overall model is huge the calculation time will be small.

IMPLEMENTATION

Models represented with XML

Based on the informal description of the chapter "Modeling" we developed a formal description for multiresolution material flow systems. We developed a XML based description language which describes our model. In addition to the simulation model, the description also contains a lot of information for the visualization during the simulation run. Each position has a location and a link to a .x file, which contains a 3D representation (mesh and texture) of the position (e.g. a 3D model of a machine). The code example (Figure 4) gives a brief expression for a position called BL. The first lines describe the localization. Palette.x is the .x file which contains the 3D Model and potential link to a texture file. Delta is the beginning of the rule for processing the events occurring at the position BL.

```
…
<Position Name="BL">
        <PX> -7 </PX>
        <PY> 0</PY>
        <PZ> 0</PZ>
        <F>palette.x</F>
        <Delta>
            <Type>In</Type>
….
```

Figure 4: XML-Sample

More detailed models are directly included in the description. The XML syntax allows to follow the hierarchical multiresolution approach with an recursive notification. E.g. the more detailed model of a position called L consists of two positions called BL and HL. This sub-model can be modeled as a traditional single resolution model and be included in the notification (see Figure 5). If these positions also include more detailed models, an hierarchy will be modeled implicitly.

```
<Position Name="L">
    <PX> 0 </PX>
    <PY> 0 </PY>
    …
    <Positions>
        <Position Name="BL">
            <PX> -7 </PX>
            <PY> 0</PY>
            …
        </Position>
        <Position Name="HL">
            <Name>HL</Name>
            <PX> 7 </PX>
            …
        </Position>
    </Positions>
</Position>
```

Figure 5: Notation of a position with a more detailed model

Execution of the software

The simulation software d[3]FACT INSIGHT needs an xml description as described above as input. A visual model is generated automatically from the description in the model.

After starting the simulation the user can walk freely through the simulation. The number of tokens assigned to the position is visualized with little red cubes. During the execution of the simulation the assignment is constantly changing. So the user can analyze the simulation within a virtual environment.

For activating more or less detailed models, at the moment, the indication by distance and indication by the direction of the view are used. A mixture of 90% distance and 10% direction leads to good results during tests.

An Example

Lets assume a production process where parts arrive, put to stock, processed and then dispatched. To model this on this level of detail we need 4 positions. In this example they are called WE (arrival), L (stock), B (processing) and WA (exit). Lets assume the stock (L) and the processing (B) consist of more detailed models. The more detailed model of L should consist of the positions BL and HL and B consists out of ST and P (See Figure 6).

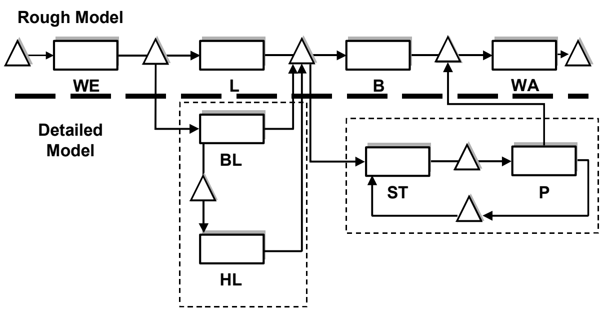

Figure 6: Model of the example process

At the beginning of the simulation run the user is standing far away from the whole model. The rough model is completely activated (see Figure 7).

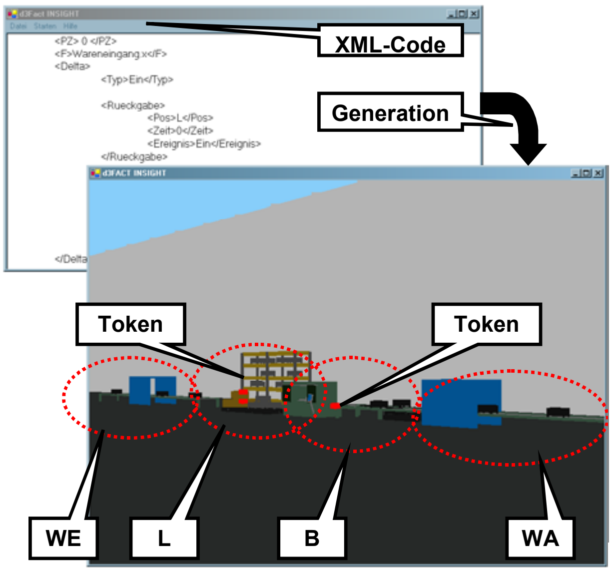

Figure 7: The user is far away. L and B are simulated with a rough model

If the user comes closer, d³FACT INSIGHT activates the more detailed models. States for the newly activated positions are generated and the 3d-representives of the more detailed positions are also activated. Figure 8 shows the situation after the activation of the most detailed level.

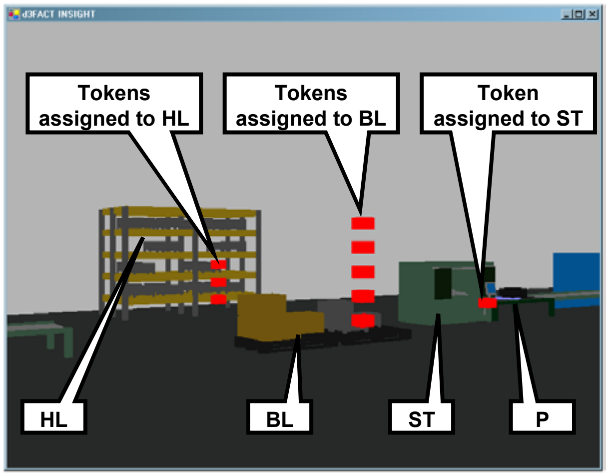

Figure 8: The User is close. The Simulation model works in high resolution.

FURTHER WORK

In the near future, the missing indications and state generation methods will be implemented. Results about the quality of the generated states are also missing at the moment.

As far as the speed up of the simulation calculation is concerned, some theoretical analysis already exists. But they need some assumptions. Most importantly, measurement with realistic models is also still needed.

CONCLUSION

Large simulations with a high resolution model cannot be analyzed interactively. But the user also cannot take care of everything at once. He only views small parts of the simulation. In our approach, only the parts the user is viewing are simulated in high resolution. Everything else is simulated with an rough model. If the user changes his interests to other areas, the high resolution area will follow him dynamically. This leads to an experience of a simulation, which is simulated completely in high resolution without the needed calculation efforts. As a compromise the calculated results are not as accurate as a complete detailed simulation.

For this we developed a modeling technique, which allows the modeler to set-up models, which can work in different levels of resolution. Rough models are not generated automatically, but the modeler has to build them. This paper demonstrates methods which can be used to analyze large models interactively in virtual environments.

REFERENCES

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

Davis, P. K. and Bigelow, J. H., 1998: Experiments in Multiresolution Modeling (MRM). RAND MR-1004-OSD,

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

Reynolds, P. F. Jr. and Natrajan, A., 1997: Consistency Maintance in Multiresolution Simulations. In: *ACM Transactions on Modeling and Computer Simulation*, 7 p. 368-392, Nr. 3, July. 1997

AUTHOR BIOGRAPHIES

Wilhelm Dangelmaier was director and head of the Department for Cooperate Plan--ning and Control at the Fraunhofer-Institute for Manufacturing. In 1990 he became Professor for Facility Planning and Production Scheduling at the University of Stuttgart. In 1991, Dr. Dangelmaier became Professor for Business Computing at the Heinz Nixdorf Institute; University of Paderborn, Germany. In the year 1996, Prof. Dangelmaier founded the Fraunhofer-Application Center for Logistics Oriented Production. His e-mail address is whd@hni.uni-paderborn.de. The web address of his working group is: wwwhni.upb.de/cim

Bengt Mueck studied computer science at the University of Paderborn, Germany. Since 1999 he is a research assistant at the group of Prof. Dangelmaier, Business Computing, esp. CIM at the Heinz Nixdorf Institute of the University of Paderborn. His main research interests are logistics systems and tools to simulate those systems in different levels of resolution. His e-mail address is mueck@hni.uni-paderborn.de.

Christoph Laroque studied business computing at the University of Paderborn. Since 2003 he is PhD student in the graduate school of dynamic intelligent systems and research assistant at the group of Prof. Dangelmaier, Business Computing, esp. CIM. He is mainly interested in material flow simulation and the "digital factory". His e-mail address is laro@hni.uni-paderborn.de.

Kiran Mahajan studied Mechanical Engineering (Specialization Production Engineering and Organization) at the Delft University of Technology. Since 2004 he is PhD student at the graduate school of dynamic intelligent systems and research assistant at the group of Prof. Dangelmaier, Business Computing, esp. CIM. His research interests include optimization of manufacturing systems using innovative simulation technologies. His e-mail address is kiran@hni.uni-paderborn.de.