Improved Process Planning by a Material Flow Simulation with Multi-User-Support

Wilhelm Dangelmaier
Christoph Laroque
Bengt Mueck
Peter Renner

Heinz Nixdorf Institute, University of Paderborn
Fürstenallee 11
D-33102 Paderborn, Germany
E-mail: {whd|laro|mueck|renner}@hni.upb.de

INTRODUCTION

Large simulation processes are typically organized by a team. Especially the planning of new processes requires a lot of different skills. Simulation and visualization are established methods supporting these tasks. But especially simulation models can't be build cooperatively by a team. Until today one person has to model the whole process and perform all the simulation studies.

This paper analyses the interaction techniques, which are required to build and execute a material flow simulation model. Based on these techniques potential conflicts, aroused by parallel action of different users on one model, will be analysed. A locking, versioning and cloning based method set will be introduced and discussed to solve these problems. Finally the architecture, on which these methods work will be presented.

STATE OF THE ART

Methods of cooperation

Cooperation serves the acquisition of potentials to generate higher benefit. Enterprises with a huge amount of divisions need to coordinate the different points of view. In order to improve efficiency, methods of cooperation have to be realised. Characteristic for cooperation is the harmonization or the collective gratification of operational tasks by several independent departments.

Putting cooperation in a more global context, it must be connected closely to the terms of communication and coordination. Collaborative work is the summation of task based activities, which are done by group members to achieve goal oriented tasks and therefore gain group targets. Therefore group processes need to be initialised as communication, coordination and cooperation, which can be organised as shown in Figure 1. While communication is the understanding of multiple persons among each other, coordination is communication, which is necessary to adjust task based activities performed within the scope of collaborative work (Teufel et al. 1995).

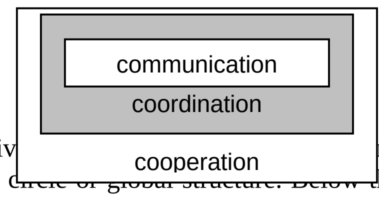

Figure 1 Group processes in collaborative work

Weidner div … ructures as the star, chain, … they are briefly shown (cp. Figure 2). So coordination adjusts local actions and decisions to fulfil global targets. For this reason coordination enables exact resource-input and efficient teamwork.

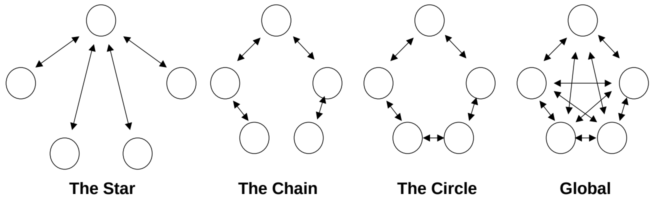

Figure 2 Communication systems (Weidner 1998)

Cooperation must also be illuminated from the perspective of computer supported collaborative work (CSCW). Collaborative work includes two essential aspects: Space and Time. Spatial viewed, collaborative work can be arranged face-to-face or in a distributed group. Temporally focused it can be arranged synchronously or asynchronously. This leads to a contemplation of collaborative work by building a space-time matrix (cp. Figure 3).

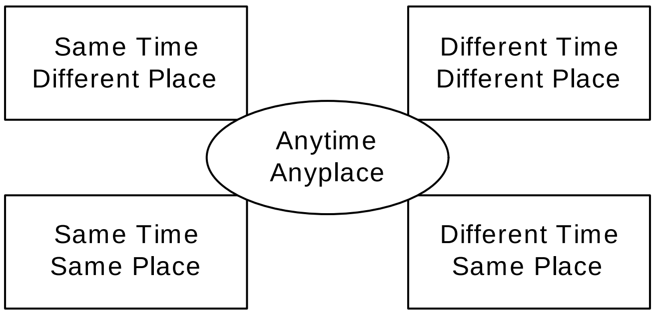

Figure 3 Anytime-Anyplace matrix (cp. Johansen 1991)

The realisation of cooperation at any time and any place can be supported by two mechanisms of communication: direct and indirect communication. While direct communication deals with propagation and management of message streams between the involved co-workers, concerning services as creation, transfer, synchronisation and filtering of message streams, indirect communication is based on a central or distributed workspace to work on shared artefacts (Schlichter et al. 1998). This is accomplished by group editors which allow a simultaneous editing and a synchronous observing of changes. Features of such group editors are part of the functionality of a single-user editor and in addition collaborative awareness, concurrency control, replication management, versioning and locking mechanisms (Prakash 1999).

Collaborative awareness yields an understanding of the activity of others to provide a context for own activities. This is one essential assumption to work on shared artefacts.

Concurrency control is needed to ensure consistency of shared and edited data. It is cut into an pessimistic and optimistic approach. The pessimistic approach makes high demand on consistency realised by central or peripheral control. The optimistic approach does not assure consistency and in turn allows inconsistent access on shared data which is in some cases even beneficial (Davidson 1984).

Fundamentally, replication management can increase availability. Here the disadvantages of replication have to be mentioned, if co-workers have different rights to work on shared data. Access then has to be separated into writing and reading access. Regarding the writing access, the power of a distributed CSCW-system might be reduced, because the constancy of the extrapolation of logical data raises with the amount of its physical copies.

To reduce the impact of extrapolation interdependencies concerning messages have to be regarded. Both, few messages with big volume and many messages with little volume have influence on power. To decrease this access entities of a defined size will be embodied as data blocs. If modifications are made within the data blocs the system needs access algorithms to transfer the replicated data blocs to a corporate consistent state. Meanwhile, it is not permitted to successfully have access to inconsistent data.

Versioning assigns a distinct number to a replicated data bloc, in order to compare the actuality of similar data blocs in different files.

Locking mechanisms are adopted on the data bloc level and define, whether a co-worker is qualified to modify or just to read a data bloc. A lock can prevent a concurrent access on the same data bloc within the same file and in turn still admit parallel access on different data bloc within the same file.

The entire data structure of a shared artefact can be illustrated as shown in figure 4.

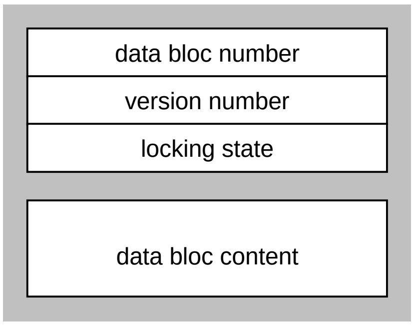

Figure 4 Structure of a data bloc (Borghoff 1998)

Cooperation in Material flow simulations

Today integrated packages like eM-Plant or Taylor ED are used for material flow simulations (Mueck and Dittmann 2003 or Klingstam and Gullander 1999). With these tools the user is able to model and execute models of production processes in one simulation environment. The whole process of modelling, execution, analysis and modification takes place on the computer of one person. If more than one user wants to edit a model, this can be done only sequential (Dangelmaier and Mueck 2003). Only one user can work at one model at the same time.

An established approach is to operate with several sub-models, where the whole model consists of sub-divisions. Each disjoined sub-model can be modelled by another team member. To calculate the whole model, one has to integrate all sub-models into a large aggregated model. In this approach all dependencies between the sub-models must be recognised by the integrating user. If changes are required, the integrated model can't be modified by more than one user at the same time. Due to the lack of versioning, it's difficult to reconstruct the changes of other team members during the modelling process. After the integration in a entire model, it is difficult to separate the sub-models again in their building blocks, for example, if one sub-model has to be replaced by a newer version. To solve this problems of interaction, the modelling process of a material flow simulation has to be regarded closer.

MATERIALFLOW-SIMULATIONS

Models

Material flow simulation models typically consists of blocks representing the modelled process entities and marks representing factors of production (e.g. parts). If two blocks interact, they are linked by connections. During the analysis phase the marks follow these connections.

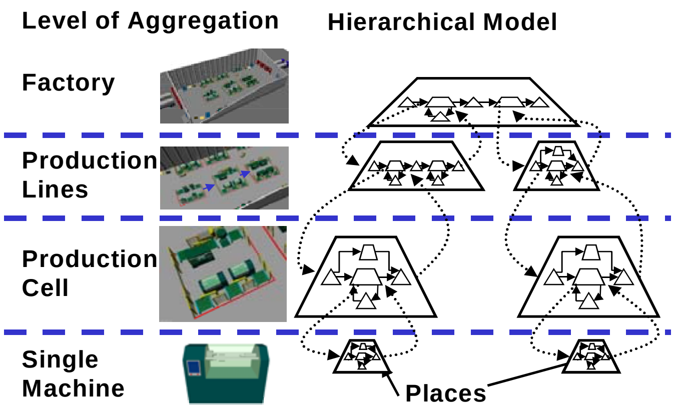

Figure 5 Hierarchical models for material flow simulation

In order to handle large models they are organized hierarchically. Several machines can be aggregated to one production cell represented as one place or block. Connected blocks can be aggregated to lines and lines can be aggregated to production stages. Links exist only within a sub-model. To interact on a more aggregated level the blocks have special connectors. The user can work with the whole aggregated building block at once. Large models become manageable (cp. Figure 5).

The Modelling Process

To Manage models and their sub-parts, several interaction techniques are required. It is necessary to allocate behaviour types to the used objects. This parameterisation defines, how objects can be placed, resp. integrated in the modelling environment in any way. Major types are: create, delete, select, move or connect objects. These types are considered closer during the next paragraph.

1. Creating objects is the fundamental step in modelling a simulation scenery. In most cases the objects are archived in component libraries. By drag & drop objects can be placed in the scenery.

2. Selecting objects is the precondition to apply other types of interaction. Selection is required before any movement on a focused object can be generated. The user needs a visual respond on the object to see, if the selection was successful. An example is given below in figure 6 (a frame which is placed around the selected object).

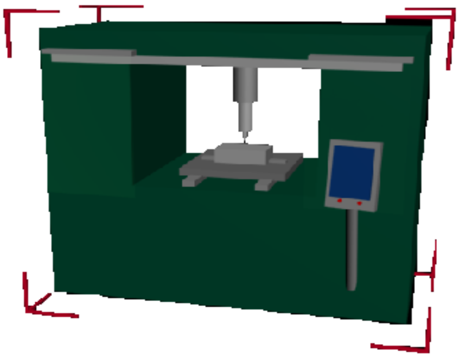

Figure 6 Selection frame around the object

3. Deleting objects is essential to make work flexible. It allows to recreate the scenery at any place and any time. Therewith exists the possibility to repair errors. It is not necessary to show the action since it is deleted anyway.

4. Referring to the layout, the movement of an object is required to organise the scenery. By moving objects a way to arrange them is given. After the selection the user can move the object within the scenery. By the use of moving features it is possible to clarify the activity structure or the hierarchical order. At best, a combination between the simulation an the 3D-factory planning is generated by placing the representatives of the simulated blocks in a most realistic environment.

5. Finally, connecting objects describes the relations between the items, e.g. how the child responses to its parents. These connections also determine the directional flow of the marks.

For every object special parameters can be adjusted, which influence its behaviour. By changing calibrations the material flow within the objects changes. Therefore a block must be selected, so that the operator has access to all specific parameters. Another aspect is the modification of the mark running through the object. Through the transformation within a block or object, it can change its condition. Modifications may be the dimension or a new shape.

As long as the working environment is limited to one user at the same time, all these types can be handled easily, but through the possibility of a multi-user interface, several conflicts may be generated by the application of several types at the same time.

The Simulation Process

After the modelling phase the execution of the simulation analysis takes place. It includes the simulation parameters, the execution itself and the analysis of the result. In contrast to the general parameters in the modelling phase, these input parameters determine the simulation flow and contain, how flexible the execution of the simulation flow can be configured and of what complexity the information about the simulation flow can be collected. During the execution the user-friendliness takes centre stage, e.g. how easy control commands can be entered (Mueck and Dittmann 2003). Several interaction techniques have to be implemented to change input data online during the execution of a simulation experiment. The analysis comprises, if there is a valid examination or an error statistic and how results can be illustrated appropriately.

Similar to the modelling phase the interaction of more than one user has influence on the execution. The next paragraph elaborates possible conflicts which might occur in a multi-user simulation environment.

CONFLICTS

The system follows the ideal of a multi-user system to model and simulate parallel in any situation. Upcoming conflicts can be separated in three cases. First, two users want to change the simulation model at the same time. Secondly, two users would like to run their simulation experiments at the same time, based on the same model. Thirdly, a user wants to change the model while another one runs a simulation experiment on the same model.

The first aspect includes the requirement for time parallel modelling since two or more persons have significant ideas or defaults to be directly integrated in the model. Possible conflicts are, for example, the access on the same object or the delete of an object, that is necessary for another.

Referring the second category of conflicts, the same simulation model is used for two or even more simulation experiments at the same time. Solving this conflict leads to another additional feature of simulation experiments. If it’s possible to simulate different parameterisations on the same model, one user can deposit several adjustments, which can be simulated at the same time on different machines.

The third aspect bases on the conclusions of the first and second one. While the simulation is running an adjusted experiment, a change in the origin model would adulterate the experiment result, because it could possibly lead to variances of the material flow concerning the modified elements running throughout the system.

Concerning all three dimensions, the next paragraph will point out the most important methods to solve the conflicts.

APPROACH

In order to handle all the described conflicts three methods have been adopted: Locking, Cloning and Versioning.

One possible way to work parallel on a subject is cloning, the multi-instantiation of a basis model. At the beginning of the working progress a clone of the actual model is created for every request. The difficulty is the integration of the different clones at the end of every process. Versioning helps to solve these conflicts, as long as it is possible to generate former versions of the actual model and compare them to the actual changes. Mechanisms have been identified , which handle those change conflicts.

During the modelling phase another method is more encouraging. By the use of lock mechanisms, several users can work on the same model. As long as they work in different parts of the model, no conflicts are generated. If two users want to access the same object, it is locked by the first user and stays locked until he finished his adjustment. Of course, a locked object has to be visualised. With available communication possibilities, the users can arrange themselves in the virtual environment.

Remembering that the model bases on different levels of hierarchies, the implementation of the locking method is more difficult. The lock of a specified object is deeply connected to the lower (more detailed) levels of the simulation model. Furthermore the upper elements connected to the locked object have to be marked, so that they also can not be changed. Figure 7 shows an example of the locking mechanism. Despite that the system allows users to work parallel on the model on each level whenever two objects are not connected in their hierarchical order. Conflicts can only occur when two different users work in the same branch. The users will directly resp. indirectly take influence on sub- or superior objects by editing sub-models of a locked node from a higher level.

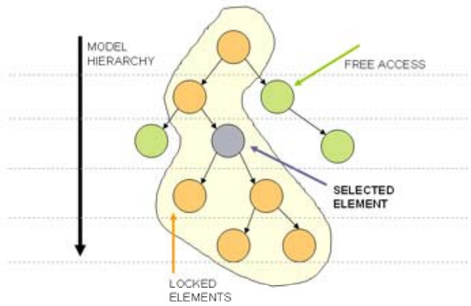

Figure 7 Locked elements in the model hierarchy

While saving the different changes of the model, several possibilities for the change management are imaginable. Beside the creation of a new version number, sub-versions are as allowed as the backup under a complete new model name. If a new (sub-)version number is given, the changes always have to be recorded, so that former versions can be computed. Every object of the basic model has its own timeline to reproduce the changes over the different versions.

Cloning is still the most easiest way, if the user wants to adjust a special simulation experiment. Through the versioning it is possible to generate older versions from the new ones, and therefore it is only necessary to save the actual version number, if a experiment is configured. The clone itself exists just over the simulation time. Afterwards the adjusted parameters and, of course, the simulation output as a compressed result resource has to be stored in the database. There is no need to archive the hole model.

The next paragraph will introduce the working architecture, in which the described mechanisms are realised.

ARCHITECTURE

The development of a entire simulation architecture for the process planning in virtual environments leads inevitably to the question, how multi-user requests to the database model are handled.

The module-based simulation environment for virtual process planning and control, which is developed at the chair of business computing, esp. CIM at the University of Paderborn, bases on a central data management of standardized objects and a controlled access to the necessary data fields. By the use of authentification and authorization the access to the data models can be limited and all applications have a restricted access to the necessary data areas. While all simulation data is held in a central storage, mechanisms of multiple access to the data areas had to be developed. Figure 8 shows the general architectural layout of the entire simulation environment.

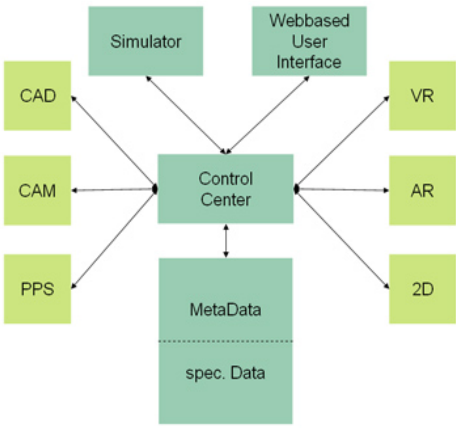

Figure 8 Architecture "digital plant”

The information flow is always attached to the central control center, where all accesses and filter methods are implemented. While every user (or application) access is bounded to the contol center, the described mechanisms for multi-user access are here integrated. The different versions are handled and administrated, and the generation of clones is initiated and controlled.

With the use of a single data area and a centralized access several modules can be established on the same data environment. Through the multi-user capabilities a parallel work on the data basis keeps possible.

CONCLUSION

Though large simulation projects can only be handled by teams, today's simulation applications don't support any multi-user capabilities.

Enabling team work in virtual environments, communication methods must be supported as well as the coordination of the collective work through coordination methods. According to the planning and control of manufacturing processes especially the development of simulation models and the parameterisation of simulation experiments must be coordinated.

With the present approach the multi-user capability of the module-based simulation environment is supported through the use of three methods: Locking, Versioning and Cloning. Whereas locking and versioning allow parallel development work on the same model, the combination of versioning and cloning leads to multiple simulation experiments based on an actual process model. The cloned model itself has not to be stored separately after the termination of the simulation process, because every version of a simulation model can be computed. The simulation input and output data are saved separately in the database system.

The system architecture of the module-based environment supports the usage of this regulating methods by the use of a central control center, which is bounded to every application relating on the consistent data area.

REFERENCES

Borghoff, U. and Schlichter, J. H. 1998, „ Rechnergestützte Gruppenarbeit“, 2. edition, pages 222-226. Springer, 1998

Dangelmaier W. and Mueck, B. 2003. “Simulation in Business Administration and Management - Simulation of production processes”. In Mohammad S. Obaidat and Georgios I. Papadimitriou, editors, *Applied System Simulation - Methodologies and Applications*, pages 382-396. Kluwer Academic Publishers, 2003

Davidson, S. B., 184, “Optimism and Consistency in Partitioned Distributed Database Systems. In *ACM Transactions on Database Systems* 9:3, pages 456-481

Johansen, R., “ Teams for tomorrow”. In Proc. 24th IEEE Hawaii Intl Conf. On System Science, pages 520-534. IEEE Comp. Soc. Press, Los Alamitos, 1991

Klingstam, P. and Gullander, P. 1999, “Overview of simulation tools for computer aided production engineering”. In: *Computers in Industry*, No. 38, pages 173-186, Elsevier, 1999

Mueck, B and Dittmann, N. 2003. *Marktanalyse Materialfluss-Simulatoren*, No. 11, ALB-HNI Verlagsschriftenreihe, Paderborn, 2003

Prakash, A., 1999, “ Group Editors”. In Beaudouin-Lafon, M., editor, *Computer Supported Co-Operative Work*, pages 103-104. John Wiley & Sons Ltd., Chichester, 1999

Schlichter, J., Koch, M. and Bürger, M., 1998, „ Workspace Awareness for Distributed Teams“. In Conen, W. and Neumann, G., editors, *Coordination technology for collaborative work*, pages 205-207. Springer, 1998

Teufel, S., Sauter, C., Mühlherr, T. and Bauknecht, K. 1995, „ Computerunterstützung für die Gruppenarbeit“, pages 9-22. Addison-Wesley, Bonn, 1995

BIBLIOGRAPHIE

WILHELM DANGELMAIER was director and head of the Department for Cooperate Planning 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, Prof. Dangelmaier has become Professor for Business computing at the HEINZ NIXDORF INSTITUTE; University of Paderborn, Germany. 1996, he founded the Fraunhofer-Anwendungszentrum für Logistikorientierte Betriebswirtschaft (ALB).

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”.

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. His main research interests are logistics systems and tools for their simulation and visualization.

PETER RENNER was born 1975 in Roermond, the Netherlands. He studied Industrial Engineering and Management at the University of Paderborn, Germany. His main research interests are virtual worlds and their application for material flow simulations.