Proceedings of the 2007 Winter Simulation Conference
S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds.

SIMULATION FRAMEWORK FOR COMPLEX MANUFACTURING SYSTEMS WITH AUTOMATED MATERIAL HANDLING

Rene Driessel
Lars Mönch

Chair of Enterprise-wide Software Systems
Dept. of Mathematics and Computer Science
University of Hagen
58097 Hagen, GERMANY

ABSTRACT

In this paper, we suggest a framework that allows for the simulation-based performance assessment of complex manufacturing systems with Automated Material Handling Systems (AMHS). Therefore, we consider a coupling architecture that connects simulation models of the manufacturing base system and the AMHS with a shop-floor control system. The center point of this architecture is a blackboard-type data layer between the shop-floor control system and the two simulation engines. We provide detailed information on how the different subsystems communicate and how each system triggers events of the other systems. We show by means of a case study how this framework supports the required performance assessment.

1 INTRODUCTION

Semiconductor manufacturing is at the heart of the electronics industry. The wafer fabrication part of semiconductor manufacturing is very complex, consisting of hundreds of process steps, diversity of product mix, re-entrant flows, sequence dependent set-ups, and batch processing (Uzsoy et al. 1992, Pfund et al. 2006). Here, batching means that we can process several lots on the same machine at the same point of time. Semiconductor wafer fabrication facilities are examples for complex manufacturing systems.

An increased level of full-factory automation is typical for 300-mm wafer fabrication facilities (wafer fabs). Therefore, an AMHS is very important for these modern wafer fabs (cf. Agrawal and Heragu 2006 and Montoya-Torres 2006 for recent survey papers on AMHS in semiconductor manufacturing) mainly because it is difficult for human operators to carry the wafers with a front-opening unified pod (FOUP). There are two types of AMHS in wafer fabs (cf. Lin et al. 2001). The first one is the interbay system that transports wafers between process bays. The second one is the intrabay system. It is used to transport FOUPS within one process bay. It is also called tool-to-tool AMHS.

Beside lot dispatching decisions vehicle routing and dispatching decisions are necessary in 300-mm wafer fabs. The basic problem for dispatching consists in selecting a vehicle from the set of idle vehicle to pickup a lot according to a certain dispatching rule. Lot scheduling and AMHS decisions usually are made independently. In (Qu et al. 2004) they suggest to incorporate AMHS decisions within a shifting bottleneck type scheduling approach. However, only a few static test instances are considered.

Therefore, it seems that lot dispatching and scheduling and vehicle dispatching and scheduling problems are rarely discussed in an integrated manner in the literature. Discrete-event simulation is an appropriate tool to study these type of problems because simulation is able to capture of the dynamic nature of the shop-floor. Lin et al. (2001) use the simulation engine SIMPLE++ to assess the performance of certain dispatching rules for vehicles. A rather sophisticated blended dispatching rule for Automated Guided Vehicles (AGV's) is suggested by Jeong and Randhawa (2001). They implement the dispatching rule within the simulation engine ARENA. Jimenez et al. (2002) use SLAM to assess the performance of several heuristics for lot and vehicle dispatching. Liao and Wang (2004) assess the performance of a neural-network-based delivery time estimation approach for automatic material handling operations by an eM-Plant simulation model. All discussed approaches do not build in any separation between the dispatching and scheduling approaches and the simulation software. Therefore, any reuse of the shop-floor control software is hard and leads consequently to a reimplementation of the algorithms.

In this paper, we extend the simulation framework of Mönch et al. (2003) to the case where additionally the simulation of an AMHS is necessary. In this particular case, we have to add an additional simulator that is responsible for the emulation of the AMHS. In this paper, we show how the suggested framework can be used to study lot and vehicle dispatching simultaneously.

However, our ultimate goal consists in extending the shifting bottleneck heuristic (cf. Mason et al. 2002 and Mönch and Driessel 2005) from flexible job shops to situations where an AMHS is also included. This requires a simulation framework as a prerequisite to allow for embedding the shifting bottleneck heuristic in a rolling horizon scheme.

The paper is organized as follows. In the next section, we discuss design criteria for the framework. Then we continue with a description of the overall framework structure including a detailed discussion of the components of the framework. In the fourth section, we describe a software prototype based on the suggested framework that supports the performance assessment of different shop-floor control algorithms. Finally, we present results of computational experiments.

2 DESIGN CRITERIA FOR THE FRAMEWORK

In a real wafer fab, a tool for dispatching and scheduling obtains its information via a message bus from the manufacturing execution system (MES) and other operative information systems, stores these information in data bases, and computes control instructions based on these information. It stores the control information in a data base for evaluation purposes. Furthermore, the MES communicates with the AMHS to provide information on delivery locations. The AMHS control system determines which vehicle has to sent to which location based on the obtained delivery location information. Finally, the dispatching and scheduling tool transmits the control instructions to the shop-floor where the instructions are executed. A control instruction is typically given by a dispatching list. The entries in this list determines which lot has to be processed next on a given machine. The dispatching list can be created by dispatching or scheduling approaches (cf. Pfund et al. 2006). A second list is necessary for vehicle dispatching. Here, an idle vehicle is selected for a given lot.

Therefore, the framework has to support the following tasks:

1. Mimic the behavior of a real shop-floor (including a base system and the AMHS) that communicates with a shop-floor control system via a message bus,
2. Provide interoperability capabilities to plug in arbitrary shop-floor control systems (i.e., scheduling and dispatching software for wafer fabs) via a blackboard-type data layer. The shop-floor control system has especially to allow for make simultaneously dispatching and scheduling decisions of lots and vehicles.
3. Allow for user-specific backups of the content of the data layer into a data base. The data base is required to reset the shop-floor control system after a machine break down in the wafer fab or to collect statistical data.

Note that parts of the design criteria 1., 2., and 3. are already design criteria of the simulation framework suggested by Mönch et al. (2003). However, the support of the AMHS is a new requirement. In this paper, we restrict our considerations on tool-to-tool AMHS.

The blackboard-type data layer is the center point of the suggested coupling architecture. It allows for a very fast access to the data required for dispatching and scheduling of lots and vehicles because the objects of the blackboard reside in the memory of the computer.

Finally, we have to discuss interoperability issues for different software systems. In our case, we have to make sure interoperability of the shop-floor control system and simulation packages. There are basically three approaches to solve this problem (cf. Mönch et al. 2003).

Text files can be used to exchange data between several applications. However, using this approach we will run into problems with synchronization. The second possible approach consists in inter process communication in the computers memory. All applications have to share memory in this case. The third approach is based on communication via network protocols. More sophisticated communication interfaces are a result of the third approach. Only a text file based coupling should be definitely avoided.

3 OVERALL FRAMEWORK STRUCTURE

In this section, we discuss the main components of the suggested architecture. Then we describe the required simulation environment. We provide some information on the design of the blackboard-type data layer in the last subsection.

3.1 Main Components of the Architecture

The framework consists of four parts that are connected together:

1. A shop floor control system that makes dispatching and scheduling decisions for lots and vehicles of the AMHS,
2. A simulation model for emulating the wafer fabrication process. It is called process model.
3. A simulation model for emulating the AMHS. It is denoted as material handling model.
4. A blackboard-type data layer that is between the two simulation models and the shop-floor control system.

Our architecture mimics the situation found in real-world wafer fabs. The two simulation models generate data that are sent to the blackboard that stores this information. A set of functions is used to replace the message bus in a real world fab. These functions allow for a clear separation of data that reside inside the two simulation models and data that is transferred to the blackboard. Note that this architecture has the advantage that we can easily split the shop-floor control system and the simulation models. The shop-floor control system needs only data that are contained in the blackboard to make lot and vehicle dispatching/scheduling decisions. No internal data from the two simulation models are going to be used for these type of decisions. The resulting dispatching lists/schedules are stored in the blackboard. They can be used as a reference data set for rescheduling activities.

Note that our design is influenced to a certain degree by the commercial simulation packages AutoSched AP and AutoMod. AutoSched AP is used to model and simulate wafer fabs, whereas AutoMod is responsible for the simulation of the AMHS. The two simulation engines are coupled by the model communication software of Brooks automation. This approach is used for example by Pillai et al. (2004) for 300-mm full-factory simulations. However, in principle we may use any other discrete-event simulation package that allows for simulating both base system and AMHS. The principle architecture is depicted in Figure 1. The separation of the entire manufacturing base system into process model and material handling system makes sense mainly from a modeling point of view.

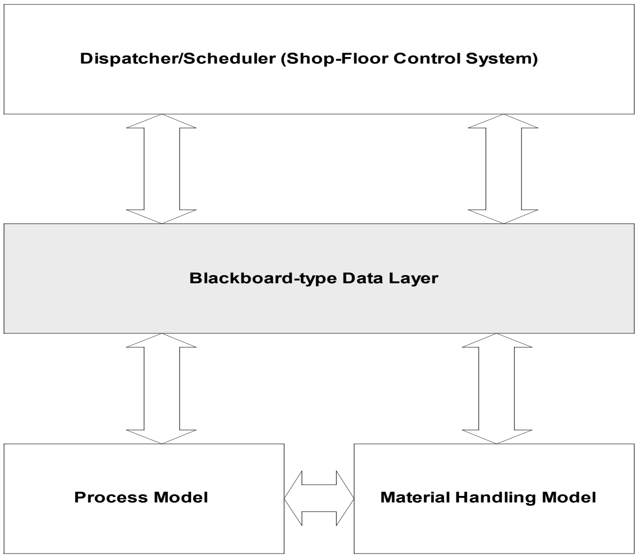

Figure 1: Overall Architecture

Usually the MES does not provide advanced production control algorithms. Therefore, out-of-the-box solutions are highly desirable. The suggested architecture allows for a seamless integration to such stand-alone dispatching and scheduling software because the functions interfacing to the simulation models can by quite easily replaced by functions interfacing to operative information systems like MES or the control system of the AMHS. The suggested design principle requires no additional changes in the software itself while allowing changes of the interface with little effort to other production planning and control applications.

3.2 Simulation Environment

The simulation environment is used to emulate the behavior of a real-world semiconductor manufacturing system including automated material handling. We are interested in assessing the performance of dispatching and scheduling algorithms for both lots and vehicles. Therefore, we have to select a simulation software that allows for model process characteristics of semiconductor manufacturing, like for example batching machines, reentrant process flows, and large size process flows. Here, we have to take into account both the process model and the material handling model.

Furthermore, we need an event-based communication between shop-floor control system and simulation models because we have to implement the decisions of the shop-floor control system into the simulation. This is basically done in the following way. When a resource (either machine or vehicle) has to make a decision (which lot has to choose next, which lot has to pick up) then the corresponding dispatching list is requested from the shop-floor control system. We have to establish an additional connection between process and material handling model when we want to avoid one total model. The required communication abilities are a huge problem of many simulation packages and shop-floor control software.

The shop-floor control system can be activated in an event-driven manner. When certain process conditions require a new dispatching list the shop-floor control system can compute such a list. On the other hand, a time-driven activation of the shop-floor control system is also possible. This feature is important in order to allow for the implementation of rolling horizon type scheduling approaches. One of the two simulation engines has to be the leading system in the coupling architecture that triggers the activation of the remaining subsystems of the suggested architecture.

A delay can be set for sending the results (i.e., dispatching lists) from the shop-floor control system to the machines and vehicles. This feature is important to mimic the situation that a shop-floor control algorithm needs some time to come up with a solution.

3.3 Design of the Blackboard-type Data Layer

The blackboard is used to store information required by the shop-floor control system to make dispatching and scheduling decisions for lots and vehicles. The blackboard contains classes that represent business objects and classes for abstract data structures (ADS). The ADS objects are basically container of the business objects. An object-oriented data base is appropriate in order to make the objects contained in the blackboard persistent in the sense of a snapshot logic.

Business objects that are related to the entities in the process model are already described by Mönch et al. (2003). We differentiate between static and dynamic data. The static data includes for example information on the machinery, process flows (routes), setup information, and data for calculating processing times. Lot release information, lot, tool, and setup states are examples for dynamic data.

The business objects that correspond to the entities of the material handling model can be classified in an analogous way. Static data are given by:

- Location: An instance of this class represents a single location in the wafer fab. Typically locations are the position of resources and storages in a wafer fab.
- Path: A path links two locations together. A vehicle can only travel on a path.
- Vehicle: A vehicle is used to transport a lot from one location to another.
- TransportStep: The transportation step represents the travel between two process steps through the wafer fab. When a lot is transported usually more than one vehicle is involved.

Dynamic data are given by the state of the vehicles. We may differentiate between the following

- Pickup: This activity represents the pickup of a vehicle for a specific lot.
- Movement: The movement activity represents a vehicle movement.
- SetDown: This activity represents the set down of a vehicle for a specific lot.

We depict the static and dynamic data for the material handling model as a UML class diagram in Figure 2. The activities shown in this diagram are used to construct a dispatching list for the vehicles.

The blackboard is used in four different situations:

1. Initialization of the objects in the blackboard at the beginning of the emulation (or at the beginning of using a specific dispatching or scheduling algorithm in a real wafer fab),
2. Update of the objects within the blackboard during the emulation (or during the shop-floor control algorithm is working) in an event-driven manner,
3. Reading of blackboard information by the shop-floor control component to make dispatching and scheduling decisions,
4. Writing of results of the shop-floor control system into corresponding objects of the blackboard.

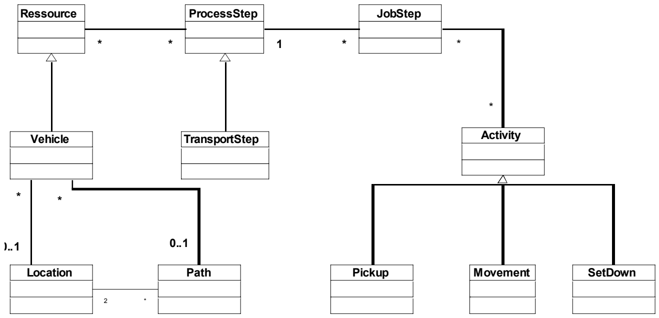

Figure 2: Class Diagram for the Material Handling Model

4 PROTOTYPE IMPLEMENTATION

The developed blackboard is an extended and refined version of the blackboard by Mönch et al. (2003) that is enriched by new classes for the AMHS domain. The blackboard is an object model that holds objects for all relevant business objects of the shop-floor.

The shop-floor control system consists for the sake of simplicity of a set of dispatching rules for the lots and for the vehicles. However, because of the design of the architecture an arbitrary shop-floor control approach is possible. We are especially interested in scheduling approaches that make lot scheduling and AMHS scheduling decisions at the same time as suggested for example by Qu et al. (2004).

The communication between the two simulation models is ensured through the Model Communication Module (MCM). A move request for a vehicle is sent from AutoSched to AutoMod for each move from one storage to another storage. The transport is simulated by the used material handling system. When a move is completed a move complete message is sent to AutoSched. The simulation within AutoSched is stopped until AutoMod has sent a corresponding complete message.

The blackboard is implemented as a dynamic link library (DLL) that is loaded by the AutoSched AP simulation engine. This DLL is written in the C++ programming language.

All necessary data structures for representing the state of the entire wafer fab are implemented within the library. The coupling with the process model is implemented by callback (notification) functions that are called for each subscribed event within the simulation.

The coupling with the material handling model is implemented by a thread (AutoModAdaptor) that implements a small complexity socket server. The AutoMod simulation engine sends a socket message to the AutoModAdapator when one of the relevant business objects changes its state.

The overall architecture including the corresponding technical details is shown in Figure 3.

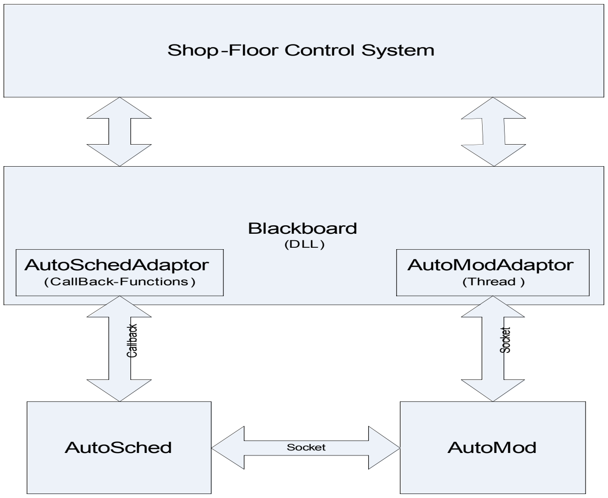

Figure 3: Implementation of the Prototype

5 CASE STUDY

We present a case study where we apply the suggested framework to the performance assessment of a batching strategy that takes future lot arrivals into account. The vehicles pick up lots of the batch with the highest priority value. This case study is an extension of previous work by the second author (Mönch and Habenicht 2003) to the situation where an AMHS is important.

We describe first the used simulation models. Then we specify the performance measures. We continue with discussing the used dispatching rules. Finally, we present the results from simulation experiments.

5.1 Used Simulation Model

We use two simulation models for our experiments. The first simulation model consists of three bays. The machinery is given by the tools of the MiniFab model (El Adl et al. 1996). The MiniFab model is freely available on the web (cf. MASM Test Data Sets 2006).

We add the AMHS described in (Shikalgar, Fronckowiak and MacNair 2002) to our simulation model. There is a bay for each group of parallel machines in the wafer fab. The bays are connected by an Overhead Hoist Transport (OHT) system. Each bay has also a OHT mono rail. Only one vehicle is used within a bay. The vehicles are not able to travel between the different bays. The layout of the material handling system is depicted in Figure 4.

The circles represent stockers that are modeled as storages in the blackboard and in the AutoSched simulator. All storages are represented as connection points within AutoMod. The dark squares are the positions of the tools within the simulation model.

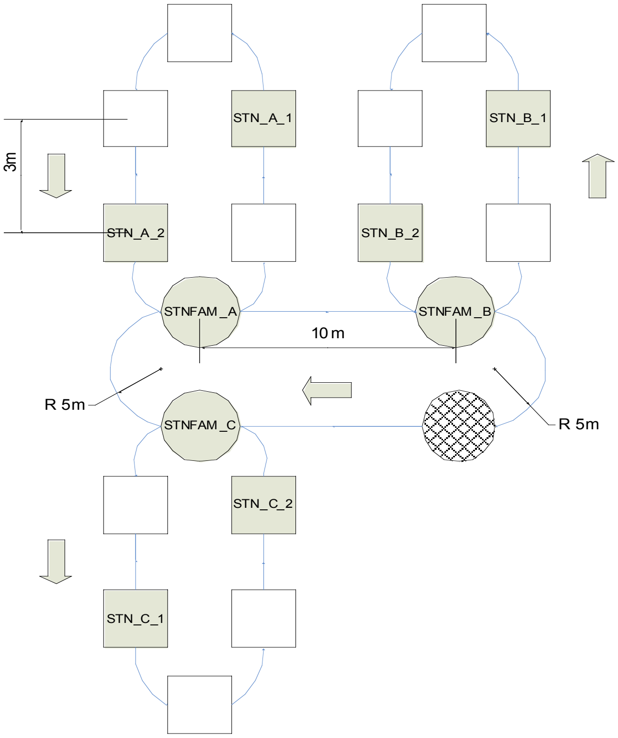

Figure 4: Layout of the AMHS for the MiniFab Model

The second model MiniFab3 is an enlarged MiniFab model. Each tool group was cloned three times. The process flows are four times longer as the original ones in the MiniFab model. Every tool group is visited by the lots between two and four times. Table 1 shows an example process flow for a product in the MiniFab3 model.

5.2 Simulation Scenarios and Performance Measures

Two load situations (moderate and high load) of the wafer fab are considered. Runs with and without the AMHS are performed for each load situation. Lots of three different product types are considered. The due dates are set using a flow factor *FF* to calculate (expected) waiting times by simply multiplying the pure processing time with the flow factor. The used flow factors and start rates of the products for the different models are shown in Table 2.

Table 1: Process Flow of product A for MiniFab3

Step	Tool Group	Step	*Tool Group*
S01	TG_A_1	S13	TG_A_1
S02	TG_A_2	S14	TG_A_2
S03	TG_A_3	S15	TG_A_3
S04	TG_A_2	S16	TG_A_2
S05	TG_A_1	S17	TG_A_1
S06	TG_A_3	S18	TG_A_3
S07	TG_B_1	S19	TG_C_1
S08	TG_B_2	S20	TG_C_2
S09	TG_B_3	S21	TG_C_3
S10	TG_B_2	S22	TG_C_2
S11	TG_B_1	S23	TG_C_1
S12	TG_B_3	S24	TG_C_3

Four different performance measures are considered. Our primary performance measure is total tardiness (*TT*). *TT* is given by summation of the tardiness value of the different lots. The tardiness of lot *j* is defined as

$$T_j := \max\left(0, C_j - d_j\right). \qquad (1)$$

Here, we denote by C_j the completion time of lot *j*. The notation d_j is used for the due date.

Table 2: Parameters of the Simulation Experiments

Product	Order repeat time for moderate load (min)	Order repeat time for high load (min)	*FF*
MinFab			
A	160	141	1.54
B	300	285	1.49
C	3000	2946	1.44
MiniFab3			
A	250	230	1.54
B	310	310	1.49
C	410	410	1.44

The second performance measure of interest is the average cycle time (*ACT*). The quantity *ACT* is given by

$$ACT = \frac{1}{n} \sum_{j=1}^{n} \left(C_j - r_j\right). \qquad (2)$$

Here r_j represents the release date for every lot *j* and *n* denotes the number of all completed lots. Furthermore, we consider the number of completed lots (throughput *TP*) and the number of late lots as additional performance measures.

5.3 Dispatching Rules for Lots and Vehicles

We use two different dispatching rules for lot dispatching. First we implement a First-In-First-Out (FIFO) rule. The lot that arrives first in the stocker of a group of machines is selected first for processing.

As a second rule we apply the Dynamic Batch Dispatching Heuristic (DBDH) (Mönch and Habenicht 2003). This heuristic takes future lot arrivals into account. Only lots of one incompatible family can be batched together. A time window $\left(t, t + \Delta t\right)$ is considered. The lots of family *i* that are ready for processing at time *t* or that will arrive within the time window are denoted by

$$M\left(i, t, \Delta t\right) := \left\{ j \mid r_{ij} - t + \Delta t \right\}. \qquad (3)$$

Here, the notation r_{ij} is used for the ready time of lot *j* of family *i*. These information can be obtained either from a upper lot planning approach or from the MES (cf. Mönch and Habenicht 2003 for more details). For Δt we choose 50% of the average processing time of the lots queuing in front of the group of parallel machines.

The elements of $M\left(i, t, \Delta t\right)$ are sorted with respect to their Apparent Tardiness Cost (ATC) (cf. Pinedo 2002) index

$$I_{ij,dyn}\left(t\right) = \frac{w_{ij}}{p_j} e^{-\frac{\left(d_{ij} - p_j - t + \left(r_{ij} - t\right)^+\right)^+}{k\bar{p}}} \qquad (4)$$

in descending order. The notation $x^+ := max\left(x, 0\right)$ is used for abbreviation. In expression (4), the notation p_j is used for the processing time of lot *j* of family *i*, d_{ij} is used for the due date of lot *j* of family *i*, w_{ij} for the corresponding weight, k is a look-ahead parameter (cf. Mönch and Habenicht 2003 for its appropriate selection), and $\bar{p}$ is the average processing time of the queuing lots. From this list for the first #lots all batch combinations are formed. For each formed batch *bj* we calculate the batch ATC index

$$I_{bj}\left(t\right) = \frac{w_{bj}}{p_j} e^{-\frac{\left(d_{bj} - p_j - t + \left(r_{bj} - t\right)^+\right)^+}{k\bar{p}}} \frac{n_{bj}}{B}. \qquad (5)$$

Here, we denote by:

- $d_{bj} := \min_{j \in B_{bj}} \left(d_{ij} \right)$: minimum due date among the lots contained in batch bj ,
- $r_{bj} := \min_{j \in B_{bj}} \left(r_{ij} \right)$: maximum ready time of the lots in batch bj ,
- w_{bj} : average weight of the lots that form batch bj ,
- n_{bj} : number of lots in batch bj ,
- B: capacity of a batch.

We use $w_{bj} = 1$ in our experiments. The idea behind this rule is that it is sometimes more advantageous to leave a machine idle and wait for an important lot instead of processing a full batch with unimportant lots.

For the dispatching and scheduling of the vehicles of the AMHS we consider three strategies. First, we take a nearest lot first (NLF) approach. The lot that is at the nearest location of the current vehicle will be transported.

As a second approach the earliest due date (EDD) rule is considered. The lot with minimum due date is first transported.

The last approach is more sophisticated. The waiting lots are sorted in such a way that lots of a formed batch will be processed together. We apply this rule to all machines. When comparing two batches the batch with the earliest calculated start date and the batch with the highest calculated ATC index is preferred. This rules is called Aligning Dispatching Heuristic (ADH) because it is strictly based on batches that are formed by DBDH. Note that type of rules cannot be implemented without the information that are stored in the blackboard.

5.4 Results of Computational Experiments

For each simulation scenario (high and moderate load, with and without AMHS) experiments with all dispatching rules are run for 20 days simulation time. Table 3 shows the results of six simulation experiments with the MiniFab model. The lots are dispatched using the FIFO rule. Table 4 shows the same results for the MiniFab3 model.

We see that *TT* and *ACT* typically increases when the material handling system is added to the models. The results for *TT* and *ACT* can be slightly improved using the due date oriented EDD dispatching rule for the vehicles. EDD is not able to guaranty that the formed batches for the tools are transported in the right order. This might lead to a poor performance in highly loaded systems.

The time to perform a simulation experiment without automated material handling is approximately 20 seconds for the MiniFab model and 15 minutes for the MiniFab3 model. It increases up to 20 minutes respectively 16 hours when the AMHS is added. We are currently working to reduce the simulation time.

Table 3: Results for FIFO Lot Dispatching Rule (MiniFab)

Experiment	**TT (h)**	**ACT (h)**	**TP**	**#Late Lots**
Moderate load				
No AMHS	79:33	15:12	279	95
AMHS, NLF	123:03	15:32	277	119
AMHS, EDD	118:32	15:30	277	116
High Load				
No AMHS	162:53	15:46	305	152
AMHS, NLF	186:02	15:56	306	164
AMHS, EDD	176:42	15:53	306	162

Table 4: Results for FIFO (MiniFab3)

Experiment	**TT (h)**	**ACT (h)**	**TP**	**#Late Lots**
Moderate load				
No AMHS	12653	60:58	279	279
AMHS, NLF	15059	71:11	271	271
AMHS, EDD	14592	69:27	271	271
High Load				
No AMHS	12830	60:19	287	287
AMHS, NLF	15714	71:32	281	281
AMHS, EDD	16282	80:44	250	250

Table 5 and 6 shows the simulation results for the experiments with DBDH. The vehicles are dispatched by NLF and ADH.

DBDH performs quite well also when the transport system is added. It might be surprising that sometimes the results for the moderate load situation sometimes even become better when the AMHS is added. The reason is the due date and slack orientation of DBDH. When the due dates are wide for a moderate load of the wafer fab then DBDH tends to form batches that are not appropriate.

The impact of the AMHS is also shown. The ADH improves *TT* up to 30% in a high load situation. However, because of the tight integration with DBDH it performs poor in some situations compared to NLF in the moderate load situation.

Table 5: Results for DBDH (MiniFab)

Experiment	TT (h)	ACT (h)	TP	#Late Lots
Moderate load				
No AMHS	59:07	14:19	278	50
AMHS, NLF	33:21	14:40	279	58
AMHS, ADH	52:14	14:18	280	47
High Load				
No AMHS	31:36	14:12	306	25
AMHS, NLF	72:54	14:18	278	48
AMHS, ADH	50:48	14:20	279	49

Table 6: Results for DBDH (MiniFab3)

Experiment	TT (h)	ACT (h)	TP	#Late Lots
Moderate load				
No AMHS	12130	58:47	281	281
AMHS, NLF	13503	64:43	275	275
AMHS, ADH	13458	64:33	275	275
High Load				
No AMHS	12762	59:55	288	288
AMHS, NLF	14291	66:07	283	283
AMHS, ADH	14269	65:41	285	285

6 CONCLUSIONS

In this paper, we described a simulation framework for complex manufacturing systems with AMHS. This work is an extension of previous research by the second present author. We are interested in using the simulation framework for performance assessment of full wafer fabs with AMHS. We derived several design criteria for such a framework. We described a prototype that is based on the simulation engines AutoSched AP and AutoMod. The suggested framework offers the advantage that there is a clear separation between the shop-floor control algorithms and the simulation models. Hence, we ensure reusability of the shop-floor control software because we do not need to implement the algorithms within the proprietary simulation software. On limitation of the suggested approach are the huge computational burden, especially for the AMHS part of the models

There are several directions for future research. We are interested in developing a shifting bottleneck heuristic for wafer fabs that takes AMHS operations into account. Some promising steps towards this goal are described by Qu et al. (2004). Based on the suggested framework a simulation based assessment of the performance of this scheduling heuristic in a rolling horizon setting seems to be possible. However, carrying out all the details is part of future research.

A second direction of future research is given by adding more details to the AMHS models in order to be closer to real-world scenarios. For example, is seems to be possible to consider bays with more than one vehicle and also interbay scenarios.

REFERENCES

Agrawal, G. K. and S. S. Heragu. 2006. A Survey of Automated Material Handling Systems in 300-mm Semiconductor Fabs. *IEEE Transactions on Semiconductor Manufacturing*, 19(1), 112-120.

Atherton, L. F. and R. W. Atherton. 1995. *Wafer Fabrication: Factory Performance and Analysis*. Kluwer Academic Publishers, Boston, Dordrecht, London.

El Adl, M. K., A. A. Rodriguez, and K. S. Tsakalis. 1996. Hierarchical Modeling and Control of Re-entrant Semiconductor Manufacturing Facilities. *Proceedings of the 35th Conference on Decision and Control*, Kobe, Japan.

Jimenez, J., Kim, B., Fowler, J. W., Mackulak, G., Choung Y. I., and Kim, D.-J. 2002. Operational Modeling and Simulation of an Inter-Bay AMHS in Semiconductor Wafer Fabrication. *Proceedings of the 2002 Winter Simulation Conference*, 1377-1382.

Jeong, B. H. and S. U. Randhawa. 2001. A Multi-attribute Dispatching Rule for Automated Guided Vehicle Systems. *International Journal of Production Research*, 39, 2817-2832.

Liao, D.-Y. and Wang, C.-N. 2004. Neural Network-Based Delivery Time Estimates for Prioritized 300-mm Automatic Material Handling Operations. *IEEE Transactions on Semiconductor Manufacturing*, 17(3), 324-332.

Lin, J. T., F. K. Wang, and P. Y. Yen. 2001. Simulation Analysis of Dispatching Rules for an Automated Interbay Material Handling System in Wafer Fab. *International Journal of Production Research*, 39, 1221-1238.

MASM Test Data Sets. 2006. Available via <http://www.-eas.asu.edu/~masmlab> [accessed June 05, 2007].

Mason, S. J., J. W. Fowler, and W. M. Carlyle. 2002. A Modified Shifting Bottleneck Heuristic for Minimizing Total Weighted Tardiness in Complex Job Shops. *Journal of Scheduling,* 5 (3), 247-262.

Mönch, L. and R. Driessel. 2005. A Distributed Shifting Bottleneck Heuristic for Complex Job Shops. *Computers & Industrial Engineering*, 49, 673-680.

Mönch, L. and Habenicht, I. 2003. Simulation-Based Assessment of Batching Heuristics in Semiconductor Manufacturing. *Proceedings of the 2003 Winter Simulation Conference*, 1338-1345.

Mönch, L., O. Rose, and R. Sturm. 2003. Simulation Framework for the Performance Assessment of Shop-Floor Control Systems. *SIMULATION: Transactions of the Society for Modeling and Simulation International*, 79(3), 163-170.

Montoya-Torres, J. R. 2006. A Literature Survey on the Design Approaches and Operational Issues of Automated Wafer-transport Systems for Wafer Fabs. *Production Plannng & Control*, 17(6), 648-663.

Qu, P., B. Steinmiller, and S. J. Mason. 2004. Incorporating Automated Material Handling Systems Into a Disjunctive Graph. *Proceedings of the 2004 Industrial Engineering Research Conference,* Houston.

Pfund, M., S. Mason, and J. W. Fowler. 2006. *Dispatching and Scheduling in Semiconductor Manufacturing.* Handbook of Production Scheduling, J. Herrmann, (eds.), Springer, Heidelberg, 213-241.

Pillai, D. D., Bass, E. L., Dempsey, J. C., and Yellig, E. J. 2004. 300-mm Full-Factory Simulations for 90- and 65-nm IC Manufacturing. *IEEE Transactions on Semiconductor Manufacturing*, 17(3), 292-298.

Pinedo, M. 2002. *Scheduling: Theory, Algorithms, and Systems.* Prentice Hall, Second Edition.

Shikalgar, S. T., D. Fronckowiak, and E. A. MacNair. 2002. 300mm Wafer Fabrication Line Simulation Model. *Proceedings of the 2002 Winter Simulation Conference*, ed. E. Yücesan, C.-H. Chen, J.L. Charnes, 1365-1368.

Uzsoy, R., C.-Y. Lee, and L. A. Martin-Vega. 1992. A Review of Production Planning and Scheduling Models in the Semiconductor Industry, Part I: System Characteristics, Performance Evaluation and Production Planning. *IIE Transactions on Scheduling and Logistics*, 24: 47-61.

AUTHOR BIOGRAPHIES

RENE DRIESSEL is a Ph.D. student in the Department of Mathematics and Computer Science at the FernUniversität in Hagen, Germany. He received a master's degree in information systems from the Technical University of Ilmenau. He is interested in semiconductor manufacturing, simulation, multi-agent-systems, and automated material handling systems. He worked for several years as a consultant for packaged ERP software. His email address is `<Rene.Driessel@fernuni-hagen.de>`.

LARS MÖNCH is a Professor in the Department of Mathematics and Computer Science at the FernUniversität in Hagen, Germany. He received a master's degree in applied mathematics and a Ph.D. in the same subject from the University of Göttingen, Germany. His current research interests are in simulation-based production control of semiconductor wafer fabrication facilities, applied optimization and artificial intelligence applications in manufacturing. He is a member of GI (German Chapter of the ACM), GOR (German Operations Research Society), SCS and INFORMS. His email address is `<Lars.-Moench@fernuni-hagen.de>`.