EnviroInfo 2004 (Geneva)
Sh@ring – EnviroInfo 2004
 ISBN: 28 29 30 275-3

Component-Based Integration of Discrete Event Simulation and Material Flow Analysis for Industrial Environmental Protection: A Case Study in Wafer Production

Volker Wohlgemuth[1], Bernd Page[1], Matthias Mäusbacher and Peter Staudt-Fischbach[2]

Abstract

In this paper a new approach is described that integrates discrete event simulation techniques with material flow analysis using a component-based framework based on Microsoft's COM-Technology. Therefore, we developed a framework architecture that offers basic functions for performing discrete event simulation in combination with material flow analysis. Furthermore, user specific components can easily be plugged in for extension purposes. Basic ideas and concepts as well as available functions of the framework are described. For empirical revision of this approach we created specific semiconductor components and analysed a wafer production process with regard to its environmental and economical aspects.

1. Introduction

Discrete Event Simulation is a very common technique for analysing operational production systems, but it is also widely used for planning purposes, especially in the evaluation of different design alternatives in production processes (see Law and Kelton 2000, Oakshott 1997). This is in particular important in the field of wafer production because of the huge expenses that come along with any investment in this highly sophisticated production process. Furthermore, the process of producing wafers is characterised by some special requirements with regard to a high-purity and dust-free production environment. Thus, changes within this production process have to be examined very carefully before bringing them into operational mode,

[1] Universität Hamburg, Fachbereich Informatik, Vogt-Kölln-Str. 30, D-22527 Hamburg, email: {wohlgemuth | page}@informatik.uni-hamburg.de, Internet: http://asi-www.informatik.uni-hamburg.de.

[2] Vishay Semiconductor Itzehoe GmbH, Fraunhoferstraße 1, D-25524 Itzehoe, email: peter.staudt@vishay.com, Internet: http://www.vishay.com.

especially with regard to bottleneck and quality assurance questions. Another important issue in wafer production is its impact on the environment. Along the production chain of a wafer, mainly different lacquers, acids, pure water and nitrogen are consumed, leading to special waste handling procedures and consideration of work safety regulations. Here, material flow analysis can be used to determine the environmental impacts of the production process. Thus, wafer production is a good example of a production process where both environmental and economic aspects have to be taken into consideration when planning changes within this process. In this sense, using discrete event simulation in combination with material flow analysis could be one promising holistic approach to estimate measures before they come into operation with respect to their impact on the environment and costs.

Up to now, you can mainly find two approaches in literature that combine simulation techniques with material flow analysis for environmental purposes. In the first approach (for example Mäusbacher 2003, Meyer 2003) on the one hand a material flow network is created as part of a special purpose software for material flow analysis, which represents the present state of all environmental relevant material flows along the production chain of a certain company for a reference period. On the other hand, a simulation model using a special simulation software is constructed. This model is used to show the dynamic behaviour of certain material flows and state variables within a reference period. Results and certain parameters of both models can now be passed to each other. Thus, alternative production processes can be compared with regard to their costs and material flows. Both models have to be maintained and controlled separately. However, the user has to handle two programs when performing a material flow analysis and a simulation study, but also has the flexibility to model each desired machine or process step.

In the second approach, which we presented in former papers (Wohlgemuth and Page 2000, Wohlgemuth 2001), the combination of discrete event simulation with material flow analysis is achieved by using a discrete event simulation model within a transition of a material flow network as a sub model. Here, the calculation of the simulation model is performed automatically and controlled by the software for material flow analysis. Also, the transfer of relevant parameters and results takes place automatically in the same software environment. As a result of such a coupling we receive - apart from material and energy flows - information about more dynamic economic performance indicators like throughput, average waiting times or mean service times. One conceptual disadvantage of this approach is the fact that a discrete event simulation can only be performed within a transition. Thus, the whole material flow network can not be simulated using discrete event simulation techniques, because the software implementing material flow networks normally doesn't have a discrete event simulation engine. Therefore, events that change the model's state in time cannot be fired. However, this approach seems to be slightly better than the first one, because no two models have to be maintained, and a simulation is performed under the control of the material flow network software in the same software environment.

2. Component-Based Material Flow Simulation

2.1 Basic Ideas

To combine the perspectives of material flow analysis and discrete event simulation in one software environment and without being limited to a single transition, we developed a component-based framework based on Microsoft's COM-Technology during the last three years. This framework shall offer basic functionalities to perform discrete event simulation studies with regard to the dynamical behaviour of the system under examination. For example, production systems can be examined to analyse critical points as well as bottlenecks. The framework will assist the user in performing material flow analysis without creating a new model and without changing the software environment. That means that a material input output balance, which is based on the same model that is used for compiling simulation results, can be created for the entire model as well as for single model elements. Thus, material flow analysis and discrete simulation are integrated in a software environment, which allows answering, for example, the following questions for the system under investigation:

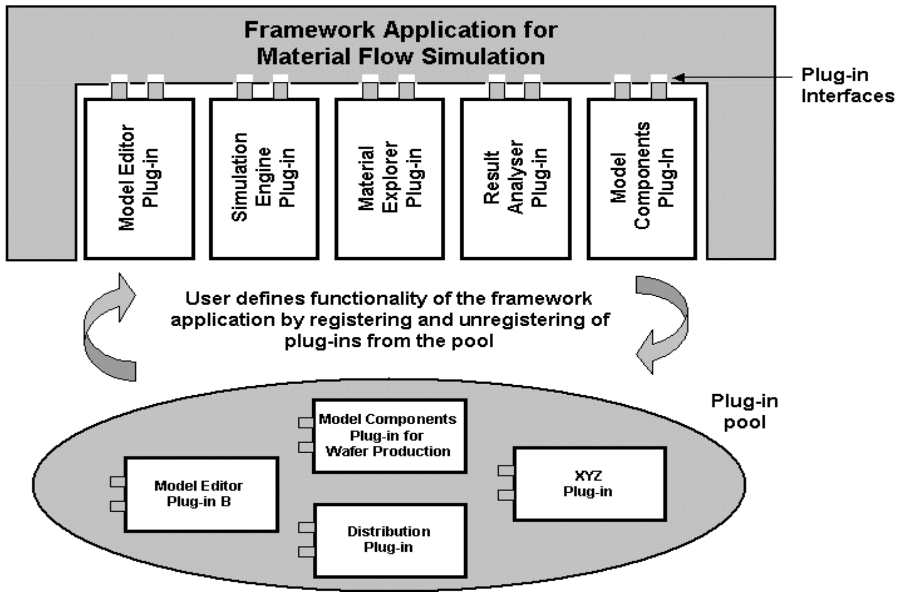
- Can we save energy if we buy a new machine without changing the throughput?
- Can we increase the utilization of our machine by changing the production process without increasing waste accumulation or energy demand?
- Can we produce more wafers without buying more lacquer and in compliance with legal limits of solvent emissions?

However, to reach this goal, we did not want to "reinvent the wheel" again. This means that we want to use as many existing functionalities as possible in our component framework. For example, it is a typical feature of existing software for material flow management to offer a material administration component where the user can define materials and their properties that are needed for a certain production process (for example, see the Industrial Environmental Management Information System Umberto®[3]). Such a material administration component should be used as a plug-in component in our framework, whenever this is possible. With regard to discrete event simulation functionality, we wanted to use a simulation engine as a plug-in that is based on a discrete event simulation framework that was developed in our working group during the last years (Page 2000). In this way, the functionality of the material flow simulator should be defined by the commands and components that are contained in the plug-ins that are registered at a framework application. Thus,

[3] See www.umberto.de.

this application shall serve as a mediator for all registered plug-ins by defining the rules of communication. Nevertheless, if the user is not satisfied with the commands and model components a certain plug-in offers, it should be possible to replace this plug-in with another plug-in that he or she has written by himself.

Figure 1 :Building a Framework Application for Material Flow Simulation consist-



ing of Component-Based Plug-ins

2.2 Basic Concepts

As mentioned before, the functionality of the framework application is created by the commands and modelling components that are registered at the framework application. Thus, the user can register and unregister certain plug-ins with the result that, if a plug-in is not registered at the framework application, its commands and components are not accessible by the user. The application organizes the commands within its menu structure or - if desired by the plug-in developer - as a button within a toolbar, which is automatically created for each plug-in. In a similar way, the framework application organizes the modelling elements that can be used to create the model in a graphical manner. Furthermore, this application creates windows in modal mode or in a dialogue fashion as a container for ActiveX-controls, if requested by a certain plug-in command (see figure 2). Finally, the framework application serves as a mediator that receives the actions of the user and transforms them into a corresponding command that is offered by a certain plug-in.

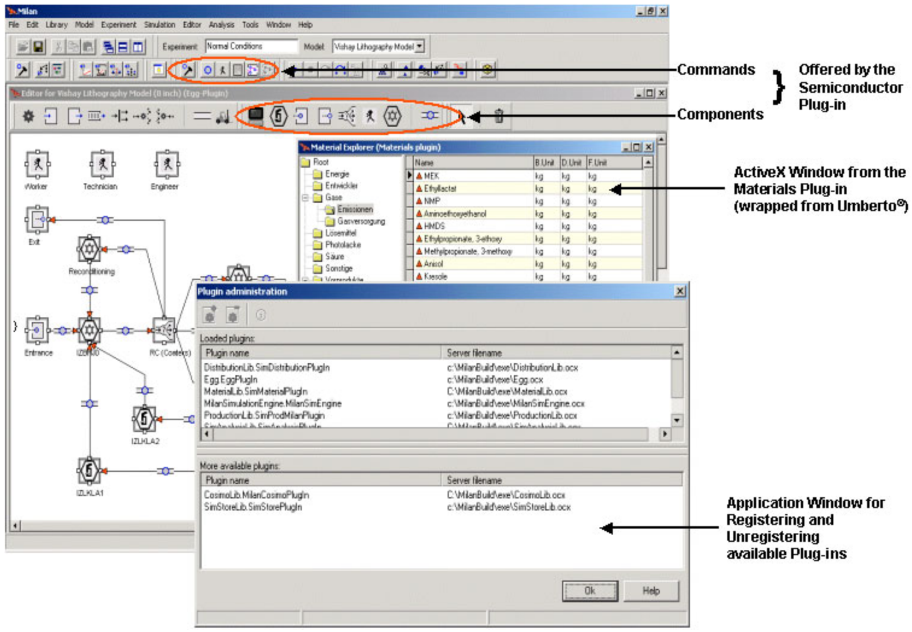


Figure 2: Commands and Components for certain Plug-ins as seen from the Framework Application

From the technical point of view, the plug-in concept is based on Microsoft's Component Object Model (COM), since the development of component-based material flow simulator started a few years ago[4]. A plug-in that wants to be registered at the framework application has to implement a certain interface, which is available from a published so-called type library that is registered at operation system level. To register a plug-in, the framework application walks through all available COM-Objects and inspects if a COM-Object has this special interface that gives access to all available commands and components. If the user accepts to register this plug-in, the framework application creates for each command and model component found in the plug-in a correspondent COM-Object that can be accessed as a button or a menu entry. Thus, all commands as well as components for creating a model are available at application level as seen in figure 2.

[4] For more details about COM see (Box 1997).

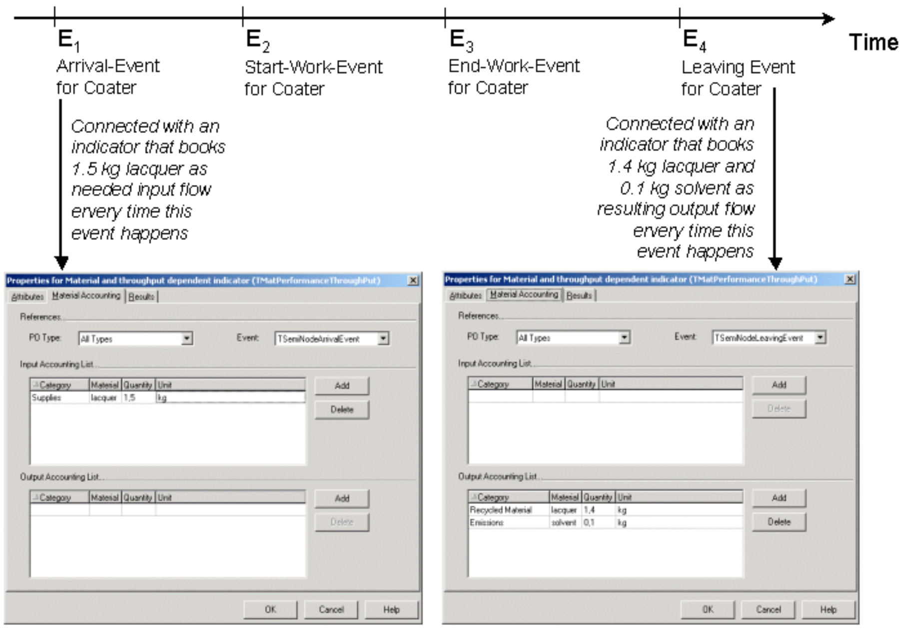


Figure 3: Relationships between Material Indicators and Events for a Coater

To get an integrated view on simulation performance indicators like throughput, utilization, adherence to delivery date etc. and material flow analysis based on the same model, we developed a concept where so-called material indicators play an important role. These material indicators combine the event-oriented worldview of our simulation engine with material bookings. A material indicator defines which materials are needed as input flows or emitted as output flows associated with a certain event or a pair of events at a specific model component. For example, you can define a material indicator for a coater that books a certain amount of lacquer as needed input flow at the start of his working process. At the end of this process less lacquer is recycled and little solvent is booked as output flow (see figure 3). Indicators are implemented within a special plug-in according to the observer pattern described in (Gamma 1998). Thus, whenever an event for a model component inside the simulation engine happens, each indicator, which is registered at this model component as an observer, decides due to the specification of the user, if a material booking has to be performed with regard to the event type or the entity that is actually handled at the model component. Furthermore, a user can develop own indicators for material bookings, if the existing indicators do not satisfy his or her needs.

In this way material input and output balances are automatically created for components of the model as well as the entire model.

2.2 Available Plug-ins

Up to now, we developed the following plug-ins for the use in the framework application in the way described above. Generally, these plug-ins can be classified into two main groups. The first group is responsible for creating discrete event simulation functionality and includes:

- A plug-in for creating random variates from a specified probability distribution (e.g. exponential, normal, uniform etc.).
- A plug-in with basic components for discrete event simulation (so-called simulation engine containing a scheduler, a clock, an event list handler etc).
- A plug-in that allows a model creation in a graphical manner (so-called model editor).
- Plug-ins for certain application fields that contain specialized model components for this application filed (e.g. model components for production systems like buffer, work stations, fork lifters, resources etc.).
- A plug-in for analysing and visualisation of simulation results (pie charts and xy-plots for single model components or for the entire model).

The second group is responsible for material flow analysis aspects and includes:

- A plug-in for the administration of all materials used in the model that is wrapped form the material flow analysis software Umberto®. Furthermore, this plug-in contains commands for the export of results and the model structure to Umberto® for further processing purposes.
- A plug-in with material indicators for the integration of discrete event simulation and material flow analysis as mentioned above.

As mentioned earlier all these plug-in components can be integrated in the application framework and can theoretically be replaced with other component-based plug-ins, if the developer of this plug-in ensures to use a few predefined interfaces. However, there are some logical dependencies that have to be considered. For example, when a user replaces the discrete event simulation engine with a simulation engine that allows continuous simulation, it does not make sense to use material indicators because they are intentionally designed to work with the discrete event simulation engine.

3. Case Study

To prove our design concepts and the applicability of such a component-based integration of discrete event simulation and material flow analysis we carried out an empirical analysis in a wafer production plant. As mentioned above the field of wafer production is a very good example where discrete event simulation as well as material flow analysis can be used because both environmental and economic aspects are of importance. Thus, one basic aim of our work was the development of a model of the wafer production process where aspects of detecting bottlenecks, planning maintenance intervals, buying new machines etc. as well as aspects of resulting emissions, needed raw materials and supplies and energy consumption can be analysed in the same software environment (our framework application Milan). Due to time constraints and for reducing model complexity we decided to analyse the lithography part of the wafer production process in a first approach.

The lithography production process plays an important role for structuring the surface of the wafer. It defines the geometry of the semiconductor components that are produced. Such a structure definition takes place by means of radiation sensitive lacquers, which are coated on the wafer, exposed using a certain mask and subsequently developed in several steps. Because this structuring happens in different layers the wafer have to run through the lithography several times depending on the semiconductor component that has to be produced. Up to now, simulation techniques and material flow analysis has not been applied to this area in the analysed company before. However, to use these techniques in this area seems to be necessary with regard to the detection of bottlenecks and in compliance with legal limits of solvent emissions.

To create a model of the lithography the basic concepts and architecture of the framework application was used to create a specific plug-in. Thus, we could focus our work on the development of specific model components, which exactly map the relevant processes to our model. For this, we developed six special components (a workstation that represents coater, stepper, developer and repacker, a component that represents a measuring devise, entry and exit points as components that represent the system border, a routing control that determines the path a wafer has to take next and a component that connects the components mentioned before). Developing these components according to the requirements of the framework and creating a model of lithography consisting of them was completed relatively fast within two month. Other functionalities like the model editor, the discrete event simulation engine and the integrated view on material flow analysis could be used from the framework without any modification. Furthermore, we had to develop some enhancements of the existing analysis plug-in with regard to the presentation of dwell times and other semiconductor specific simulation results. However, using the framework eases our effort to get an integrated view on simulation results and material flow analysis as shown in figure 4. Actually, there is still work in progress. We are now elaborating an alternative model that shows the lithography process that uses 8-inch wafer instead of 6-inch wafer because there are some tangible plans to change to this wafer type on near future.

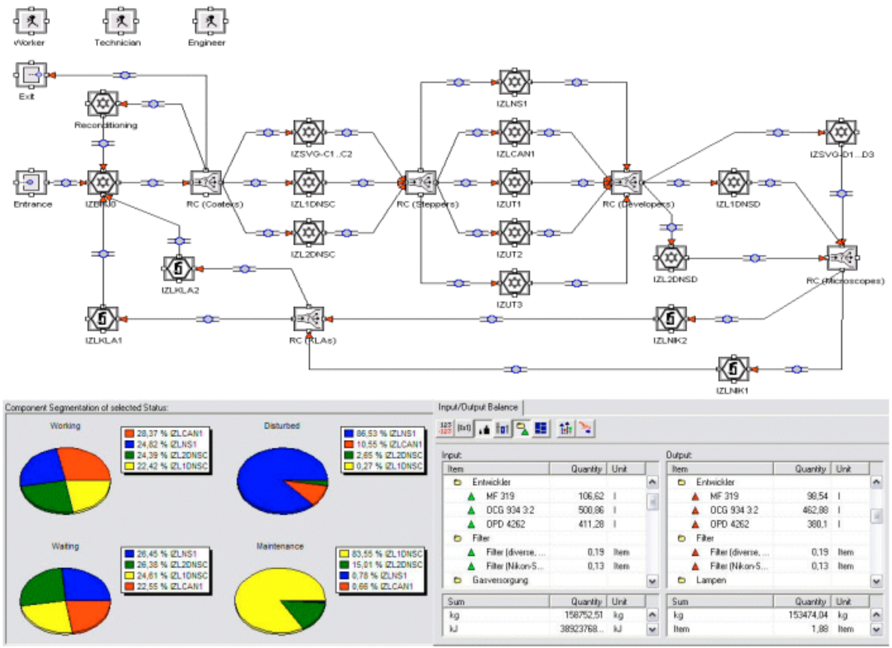


Figure 4
Model of the Lithography Area consisting of Semiconductor Specific Components and Resulting Integrated View on Simulation Results and Material Flow Analysis

4. Summary and Outlook

Discrete event simulation and material flow analysis are sophisticated techniques that require profound knowledge and are assisted by specialized software tools. Both techniques have different roots. However, they are both based on a computer model that reflects the production process under examination. Up to now, two models that are handled in two software tools are needed to perform simulation studies and material flow analysis of a certain production process. In consequence, these two models have to be maintained redundantly if a model has to be changed. In this paper we presented an approach that offers an integration of these both techniques based on one model in the same software environment. Thus, this approach reduces the effort that is needed to offer such an integrated view. Furthermore, this approach allows the user to concentrate on the development of specialized model components for a certain application field. As an example we described the application of our approach in the field of wafer production.

As a key concept of this integrated view we mentioned material flow indicators that combine material bookings with events. However, up to now material bookings that are not connected directly with a certain model component (e.g. cleaning process water that is coming from single machines and is treated in a central waste water treatment plant) cannot be taken into consideration. We will work on this subject in future.

Bibliography

Box, D. (1997): Essential COM. Addison-Wesley, Reading.

Gamma, E. (et al.) (1998): Design Patterns: Elements of reusable object-oriented software, Addison-Wesley, Reading.

Law, A.M., Kelton, W. D. (2000): Simulation Modeling and Analysis, McGraw-Hill, Boston.

Mäusbacher (et al.) (2003): Kopplung von Stoffstromnetzen mit einem Simulationsmodell der Produktionsendstufe eines Halbleiter-Herstellers. In: Wittmann, J., Dimitris, K. M. (Eds.): Simulation in Umwelt- und Geowissenschaften, pages 167-179, Shaker, Aachen.

Mayer, M. (2003): Möglichkeiten der Simulation im Rahmen des betrieblichen Stoffstrommanagements. In: Tschandl, M., Posch, A. (Eds.): Integriertes Umweltcontrolling. Von der Stoffstromanalyse zum integrierten Bewertungs- und Informationssystem, pages 43-57, Gabler, Wiesbaden.

Oakshot, L. (1997): Business Modelling and Simulation, Pitman, London.

Page, B, Lechler, T., Claassen, S. (2000): Objektorientierte Simulation in Java mit dem Framework Desmo-J. Libri Books on Demand, Norderstedt.

Wohlgemuth, V., Page, B. (2000): Einbettung von Transportmodellen und diskreten Simulationsmodellen in Stoffstromnetze. In: Cremers, A. B., Greve, K. (Eds.): Umweltinformatik `00. Umweltinformation für Planung, Politik und Öffentlichkeit. 14. Internationales Symposium "Informatik für den Umweltschutz" der Gesellschaft für Informatik (GI), Bonn 2000, Band 1, pages 343-360, Metropolis, Marburg.

Wohlgemuth, V., Bruns, L., Page, B. (2001): Simulation als Ansatz zur ökologischen und ökonomischen Planungsunterstützung im Kontext betrieblicher Umweltinformationssysteme (BUIS). In: Hilty, L. M., Gilgen., P. W. (Eds.): Umweltinformatik `01. Umweltinformation für Planung, Politik und Öffentlichkeit. 15. Internationales Symposium "Informatik für den Umweltschutz" der Gesellschaft für Informatik (GI), Zürich 2001, Band 2, pages 999-1008, Metropolis, Marburg.