

## **TEST PROBLEMS, REFERENCE MODELS AND FAB SIMULATION**

Po-Chen Lin  
Leon McGinnis

Georgia Institute of Technology  
W. M. Keck Virtual Factory Lab  
School of Industrial and Systems Engineering  
Atlanta, GA 30332-0205, USA

### **ABSTRACT**

Semiconductor fabs are among the world's most expensive factories, and present some of the most interesting manufacturing planning and control problems. It's not a surprise that simulation is the workhorse analysis methodology, both in practice and in research. Over the past thirty years, a number of "standard" wafer fab test problems have been published to provide a common basis for comparing proposed planning and control policies and algorithms. We present a reference model, expressed in an analysis application agnostic language, OMG SysML™, that can be used to specify any of these test problems. We then show how this reference model also can be used to automate the generation of analysis models for one particular simulation solver, and argue for the use of parametric problem generators as a way to more fully explore planning and control methods.

### **1 INTRODUCTION**

In the semiconductor manufacturing domain, many OR models have been proposed to support decision making, and there is a large literature addressing fab operations. The decisions examined include planning (how much of what to produce in future time periods), work release (when to start a lot into production), batching (how to assemble lots into batches for processing), and dispatching (which of the available lots to process "next"). Invariably, simulation models are used to evaluate proposed decision methods or algorithms.

Recognizing that some basis is needed for comparing results from decision methods or algorithms proposed by different researchers, a number of "standard" wafer fab test problems have been proposed. In this paper, we consider in detail the following set of test problems:

- Mini-Fab (Kempf 1994)
- MIMAC I-VI (Fowler and Robinson 1995)
- Kayton 1997 (Kayton et al. 1997)

It should be noted that these test problems specify only the base system (Mönch, Fowler, and Mason 2013). All consideration for planning and control is left to the user of the test problems.

In engineering design, there is a clear distinction between models of the artifact being designed, and analysis models that support the designer in making design decisions. In mechanical design, as an example, a solid model created in a CAD package represents the artifact. To understand how the artifact may behave under mechanical or thermal stress, a finite element model is created, analyzed, and the results displayed using the solid model of the artifact. This system model-analysis model separation allows researchers to propose, develop and test new analysis models without having to worry about the specification of the system itself. There is a standard representation of the artifact.

The "reference model" for the artifact domain (above, a mechanical part) defines the semantics and syntax for specifying an instance. For example, constructive solid geometry (CSG) is one such reference model. Conforming software tools for authoring and editing instance models can then exchange instance

models, i.e., models of specific mechanical parts. Moreover, analysis models, e.g., to perform finite element analysis of the thermal properties of the mechanical part, can be developed independently of the particular software used to create/edit the instance model, because the instance model itself conforms to a standard representation.

This is not the case for modeling and analyzing wafer fabs. The test problems are “system instances” from the wafer fab domain. These instance models are distributed as data tables, with semantics implied by record or field names. The user of the test problem typically must correctly interpret the data table names. The system analysis models are test-problem specific simulations used to test proposed planning or scheduling methods. The first goal of this paper is to demonstrate that these test problems all share a common structure that can be specified as a reference model using a standard system modeling language, OMG SysML™. One result of having such a reference model is that it defines the corresponding data schema that provides a standard way of distributing instance models.

Since these test problems address only the base system, or plant, the second goal of this paper is to elaborate the reference model to incorporate plant-control separation in specifying the test problems. Since the test problems themselves do not address control, our initial demonstration will assume some simple default controls. Clearly, these simple illustrations will not capture the full complexity of wafer fab control; rather they illustrate a potential approach to capturing this complexity.

The third goal of this paper is to demonstrate the value of a reference-model-based approach. Using one particular simulation tool, we show that a simulation component library can be constructed that corresponds to the reference model. The reference model allows the straightforward implementation of “constructors” that can read the test problem instance data tables, select the associated simulation model components, and link them correctly to create the corresponding fab simulation model. Thus, any variation of the test problem conforming to the reference model can be simulated with no additional simulation modeling effort. This establishes the possibility to develop and distribute not just standard test problems, but standard testbeds, where alternative planning and scheduling algorithms could be tested by creating their corresponding simulation components. Of course, each such testbed is specific to the simulation tool used to develop the component library. However, if all such testbeds conform to the reference model, then they should give similar results.

It is important to note that using the proposed reference model does not require adoption or use of the SysML language. Rather, the semantics and syntax specified in our SysML models are easily translated into data schema, allowing system instance models to be captured using standard database tools. Further, the system model components specified using SysML have well-defined interfaces, so the corresponding simulation components can be specified in any capable simulation language. One might ask, “Why not simply specify the reference model using a simulation language to begin with?” The answer is that every simulation language imposes its own world view, semantics and syntax, and all are different in some respect. Using any one simulation language to try to create a domain-specific reference model carries the risk that the resulting reference model is fundamentally limited.

The remainder of the paper is organized as follows. Section 2 briefly describes the test problems. Section 3 describes a set of manufacturing system modeling principles and presents a reference model expressed as SysML diagrams. Section 4 discusses the types of simulation components needed to simulate any of the test problems, using the Mathworks SimEvents language. Section 5 contains a brief discussion of a demonstration of simulation model generation using the reference model and a simulation component library. Finally, in section 6 we discuss the implication and potential future work.

## **2 THE TEST PROBLEMS AND APPLICATIONS**

There are many semiconductor wafer fab test problems available. The most often cited include MIMAC (Fowler and Robinson 1995), Mini-Fab (Kempf 1994), and Kayton1997 (Kayton et al. 1997), and they are accessible from <http://p2schedgen.fernuni-hagen.de/index.php?id=242>. These are the three we will use to

demonstrate our approach. Others, such as Sourirajan and Uzsoy (2007) and Wein24Fab (Wein 1988) also have data available.

These three test problems present key features of a semiconductor wafer fab base system—hundreds of manufacturing steps with reentrant flows. All three test problems specify the same basic structure: products with associated deterministic process plans executed by fab tools.

These test problems also provide some modeling flexibility. Users can easily change the lot arrival rates for given products and the number of units of each tool type. These kinds of changes might be needed to test the robustness of a proposed planning or control method with respect to changes in demand or resources. It is much less clear how one might examine sensitivity to the number of different products, or to alternative process routes, because it is not clear how those might be consistently defined. One important aspect of the test problems is that they do not address material handling. It might be argued that material transport in a wafer fab is sufficiently consistent that it can be ignored for the purposes of testing flow control algorithms. However, fabs also employ stockers for the temporary storage of lots that cannot move directly to their next process tool. Dealing with stockers presents challenges to the development of control algorithms.

In the past 30 years, a large number of publications dealing with semiconductor wafer fab operations addresses “testbeds”. Our goal here is not to provide a complete literature review, but to note, in Table 1, a few of the papers that address the key issues. All of these papers use their own simulation platforms with test problems listed above to evaluate the particular planning and control algorithms they proposed. The five decisions most examined in the literature include planning/scheduling (how much of what to produce in a given time bucket), work release (when to start a lot into production), batching (how to assemble lots into batches for processing), dispatching (which of the available lots to process next), and lead time estimation (when a lot should be released to meet a given due date).

<b>Planning/Scheduling</b>	Mason, Fowler, and Carlyle (2002), Mönch and Drießel (2005), Mason, Fowler, and Carlyle (2005), Barua et al. (2005), Mönch et al. (2007), Pfund et al. (2008), Jampani and Mason (2010), Yao et al. (2011), Ponsignon and Mönch (2012), Kacar, Irdem, and Uzsoy (2012), Kacar, Mönch, and Uzsoy (2013),
<b>Work Release</b>	Qi, Sivakumar, and Gershwin (2008), Qi, Sivakumar, and Gershwin (2009), Chen, Fan, and Chen (2009), Mönch et al. (2011), Ponsignon and Monch (2012)
<b>Batching</b>	Mönch and Habenicht (2003), Fowler, Mönch, and Rose (2006), Zimmermann and Mönch (2006), Mönch et al. (2007), Mönch, Fowler, and Mason (2012)
<b>Dispatching</b>	Dabbas et al. (2001), Dabbas et al. (2003), Dabbas and Fowler (2003), Scholz-Reiter and Heger (2009), Crist and Uzsoy (2011)
<b>Lead Time Estimation</b>	Asmundsson, Rardin, and Uzsoy (2006), Kacar and Uzsoy (2015), Kacar, Mönch, and Uzsoy (2016)

### 3 A REFERENCE MODEL FOR WAFER FAB TEST PROBLEMS

In this section, we discuss a modelling framework that is suitable for describing the semiconductor wafer fab test problems, and that we claim can be elaborated for describing fabs more generally. We first take MIMAC I test problem as an example to introduce a reference model, expressed in an analysis agnostic language, OMG SysML™ (OMG SysML 2015), that can be used to specify any of these test problems. The

discussion here elides many details of the reference model in the interest of brevity. Readers interested in a deeper dive should contact the authors.

In the 2009 Winter Simulation Conference, there were two papers addressing the possibility of generating simulation models directly from SysML models, (Schönherr and Rose, 2009), (McGinnis and Ustun, 2009). That early work lead to the conclusion that using SysML directly to create highly complex system models is probably not the correct approach in this domain. The approach presented here is quite different, in that SysML is used to define the semantics and syntax of system models, allowing the system models themselves to be captured in any appropriate toolset.

Semiconductor wafer fab manufacturing can be modelled using the modeling framework for discrete events logistic systems (DELS) described in Sprock (2015). In this framework, *products* are produced by *processes* (activities) using *resources* and taking place in a *facility*. In this “PPRF” modeling framework, a *task* authorizes a process to execute (to use resources) to produce a product. This is an object-oriented framework, and products, processes, resources, facilities and tasks can be nested. For example, the process “assemble product XYZ” could encapsulate a number of individual processes to accomplish the acquisition, movement and placement of purchased parts, and consequently, the product “XYZ” contains the purchased products. The resource “assembly department” could contain a number of assembly workstations. The task “produce product XYZ” could contain a number of subtasks related to obtaining the necessary parts, and moving them to the assembly workstation, as well as the “assemble product XYZ” task.

A *process plan* identifies the production processes required to produce a product, and any associated precedence relationships.

A DELS has two complementary systems—a base system, which is described using the PPRF framework, and a control system. Each of these systems has certain required functionalities. In addition to the processes required to produce products identified in Figure 1, the base system must have processes capable of executing the five canonical control decisions identified by Sprock (2015): 1) admission, 2) sequencing, 3) assignment, 4) routing, and 5) resources state change. The controller must have the functional capabilities to monitor events occurring in the base system, determine which events require control decisions, apply an appropriate control strategy, and map control decisions to executable base system processes which can execute the control capabilities.

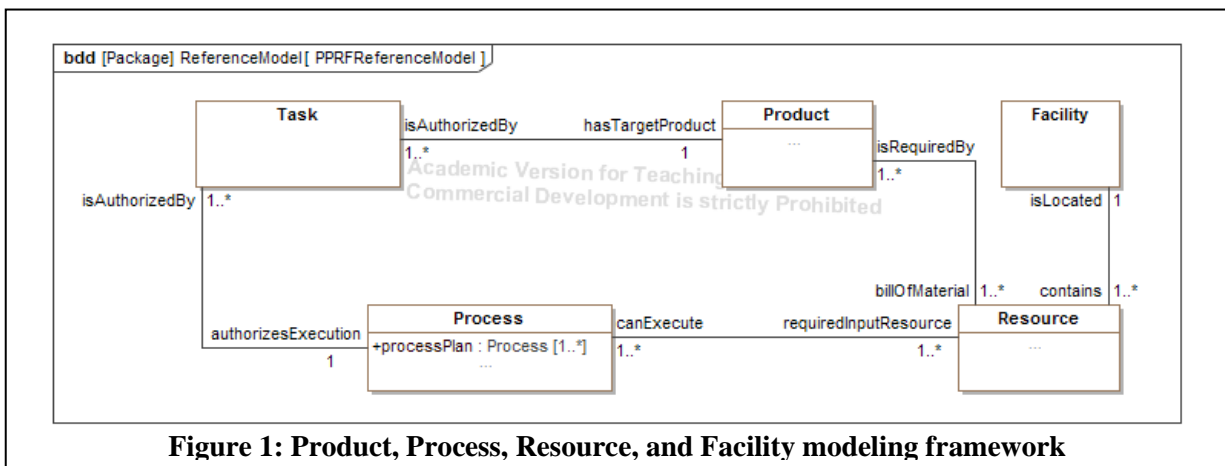


Figure 1: Product, Process, Resource, and Facility modeling framework

Figure 2 shows how the PPRF framework can be extended to model the semiconductor wafer fab test problems. Product, process, resource, and facility are extended using common wafer fab terminology, and can be given properties to represent all the test problem detailed attributes, although some of these are not shown here due to space limitations. Note that a “route” is a static specification of the sequence of production operations required for a given product. The figure also illustrates an implied control system architecture. At the fab level, there is a “route controller” to enforce the routes (process plans) specified in the test problems. ToolGroup extends from Resource to represent a collection of identical/similar resources, such as, e.g., lithography tools, diffusion furnaces, etc. There are two types of tool groups extended from ToolGroup, to distinguish, respectively, tools that process either individual wafers or lots sequentially from tools that process batches, and each tool group type has a corresponding sequencing controller.

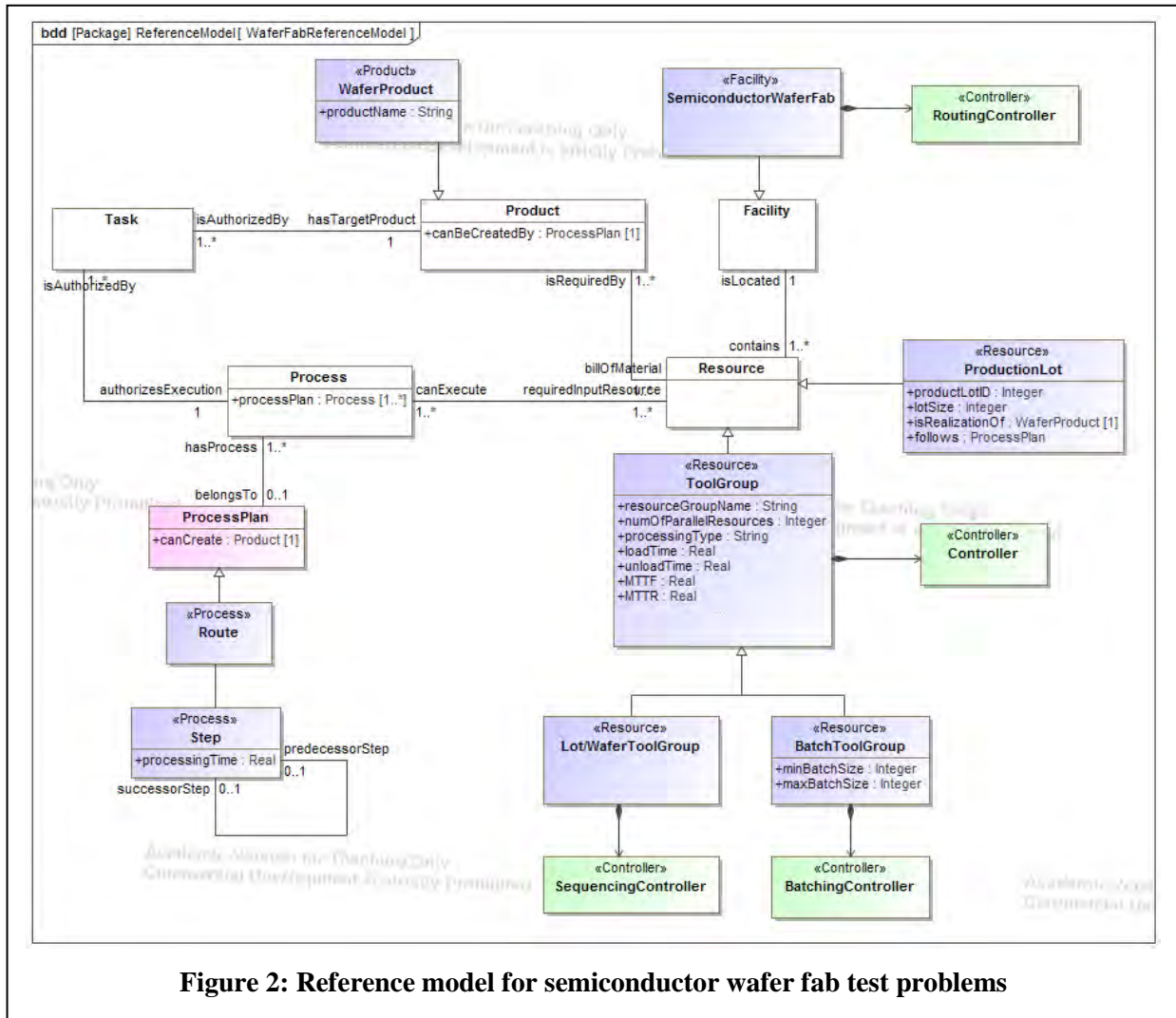


Figure 2: Reference model for semiconductor wafer fab test problems

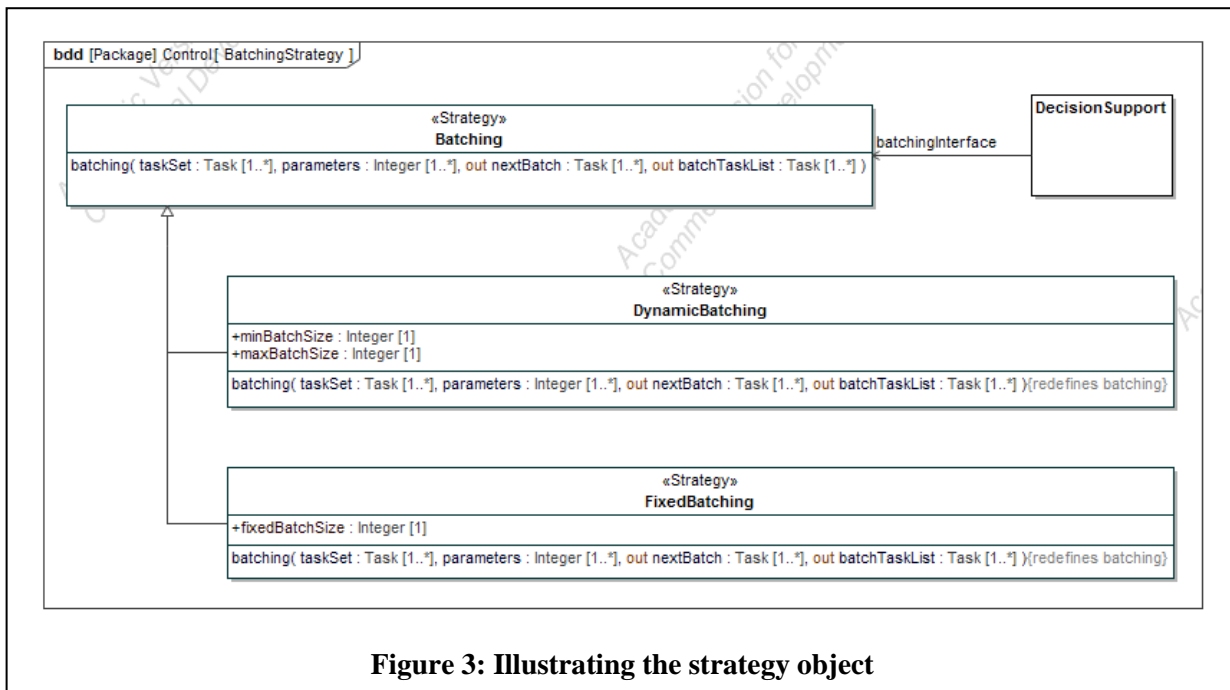
In the MIMAC I test problem dataset, there are three main data sets: data related to products (product); data related to tool groups (resource); and data related to product route in a wafer fab (process plan). Product data specifies: 1) what is the product (product name), and 2) lot size. Tool group (resource groups) data provides: 1) tool group name; 2) number of parallel tools; 3) processing types, which are lot, wafer, or batch; 4) minimum and maximum batch sizes for batch tools; 5) load and unload time; and 5) MTTF and MTTR. Product route data includes: 1) process plan of a product; 2) processing time of each step; and 3) sequencing

dependency (predecessor and successor processes). All these detailed data elements can be accommodated in a straightforward manner by the modeling framework illustrated in Figure 2.

In order to make a reference model applicable in practice, we also need to specify controls. From Figure 2, we need a routing controller to drive a lot to its correct “next” destination, based on its progress in its process route. For a lot/wafer tool group, we need a sequencing controller to determine which of the available lots/wafers to process “next”. Likewise, for a batch tool group, we need a batch controller to form and sequence batches. The test problems specify the route, but do not specify how batching and sequencing decisions are to be made. To accommodate this flexibility, the reference model employs the concept of “strategy”—for each type of decision, a generic strategy interface is defined. Thus, any strategy may be used, as long as it conforms to the interface specification. Figure 3 illustrates the strategy concept.

For example, there are two concrete strategy classes in Figure 3, *DynamicBatching* and *FixedBatching*. The batching controller can use either, depending on the control intent. With a standard interface, this abstract batching strategy class can be sub-classed and refined to support many batching solution algorithms.

The reference model illustrated in Figures 1-3 also applies to other MIMAC test problems (MIMAC II-VI) with different sets of products, process plans and tool groups. For small scale semiconductor wafer fab test problems such as Mini-Fab and Kayton1997, the reference model also applies since they share the same modelling structure with MIMAC test problems.



#### 4 SIMULATION MODELING

A fundamental insight in simulation modeling is that if we can create a simulation model that conforms to the reference model as illustrated in Figures 1-3, then perhaps we can create a set of “standard” simulation components which can be used to automate the creation of similar conforming simulation models. To demonstrate this notion, we have created a set of simulation components using the SimEvents™ language from MathWorks. SimEvents (MathWorks SimEvents 2015) is well integrated with MATLAB™, significantly simplifying the development of simulation generator code.



The semiconductor wafer fab simulation components library implemented is in MATLAB SimEvents. There are four main blocks: 1) production lot generation, 2) routing control, 3) general lot/wafer processing tool group, and 4) batch processing tool group. The routing block implements a routing controller, which was shown in Figure 2. A general lot/wafer processing tool group block is used, with appropriate parameter values, to represent any lot/wafer processing tool group. In our demonstration, the associated sequencing controller uses the default FIFO function in MATLAB SimEvents. The batch processing tool group block can represent, with appropriate parameters, any batch tool group. Since there is not a default batching function, the batch controller block implements a simple batching strategy, using the standard strategy interface. Clearly, these blocks require a significant level of coding detail, particularly for the batch controller. The interested reader will be able to examine this code at [iesystemslab.org](http://iesystemslab.org).

## **5 SIMULATION GENERATION**

In this section, the simulation generator for semiconductor wafer fab test problems is described. We have a reference model (Figure 2) to which the set of test problems conforms. We have a set of simulation modeling components (Figure 4) which conform to the reference model. The key to being able to automate the generation of test problem simulations is a mapping between the elements of the reference model which reflected in the test problems and the simulation modeling components. This mapping becomes the basis for code that can parse the test problem data and generate the corresponding simulation model.

Figure 4 shows the mapping between the reference model and the simulation modeling components. In Figure 4, the production lot (a realization of Wafer Product) in the reference model maps to the production lot generation block. The associated route (process plan) of a product maps to the attribute called “process plan” inside the task generation block. The tool group which is to execute the steps maps to the general lot/wafer tool group block or the batch tool group block in terms of what type of resource it is, batch or lot/wafer processing tool group. The routing controller in the reference model maps to the routing block. All attributes specified in the reference model associated with products, processes, resources of a test problem are also applied to this mapping between reference model and simulation component library.

We demonstrate the simulation automation with a specific use case, the MIMAC I test problem. The implementation environment is MATLAB 2015b with MATLAB SimEvents. The steps to automatically generate simulation model include:

1. Reading data of a particular test problem in terms of semiconductor wafer fab product, process, and resource reference model
2. Mapping between reference model and SimEvents components library
3. Generate simulation model of a test problem

Since we can automatically generate a simulation model for a particular conforming test problem, to make a simulation model workable, we still need to specify the control strategies to be implemented in the simulation model. For lot/wafer processing blocks, the default control is sequencing-FIFO to decide which of the available lots in the buffer to process next. For batch processing block, the default control is a rule based dynamic batching strategy with minimum and maximum batch sizes. The batch controller will query the system state and determine the next batch of lots of the same product in the same step and its batch size between maximum and minimum batch sizes specified. The last default control is the routing controller which will read the process plan, and determine if the lot just completing any step has a successor, and if so, route it to the corresponding tool group.

We use MIMAC I test problem as an example. MIMAC I has 4 products and 4 corresponding process plans, each with hundreds of steps executed by 70 tool groups. The 70 tool groups are resources and are labelled as either batch processing or lot/wafer processing tool group. Figure 5 illustrates the SimEvents model automatically generated by the steps listed above, although many of the tool groups are cropped from the figure. In Figure 5, we can see there are four different products with associated process plans and tool groups are either batch or lot/wafer processing tools that are specified by the MIMAC I test problem.

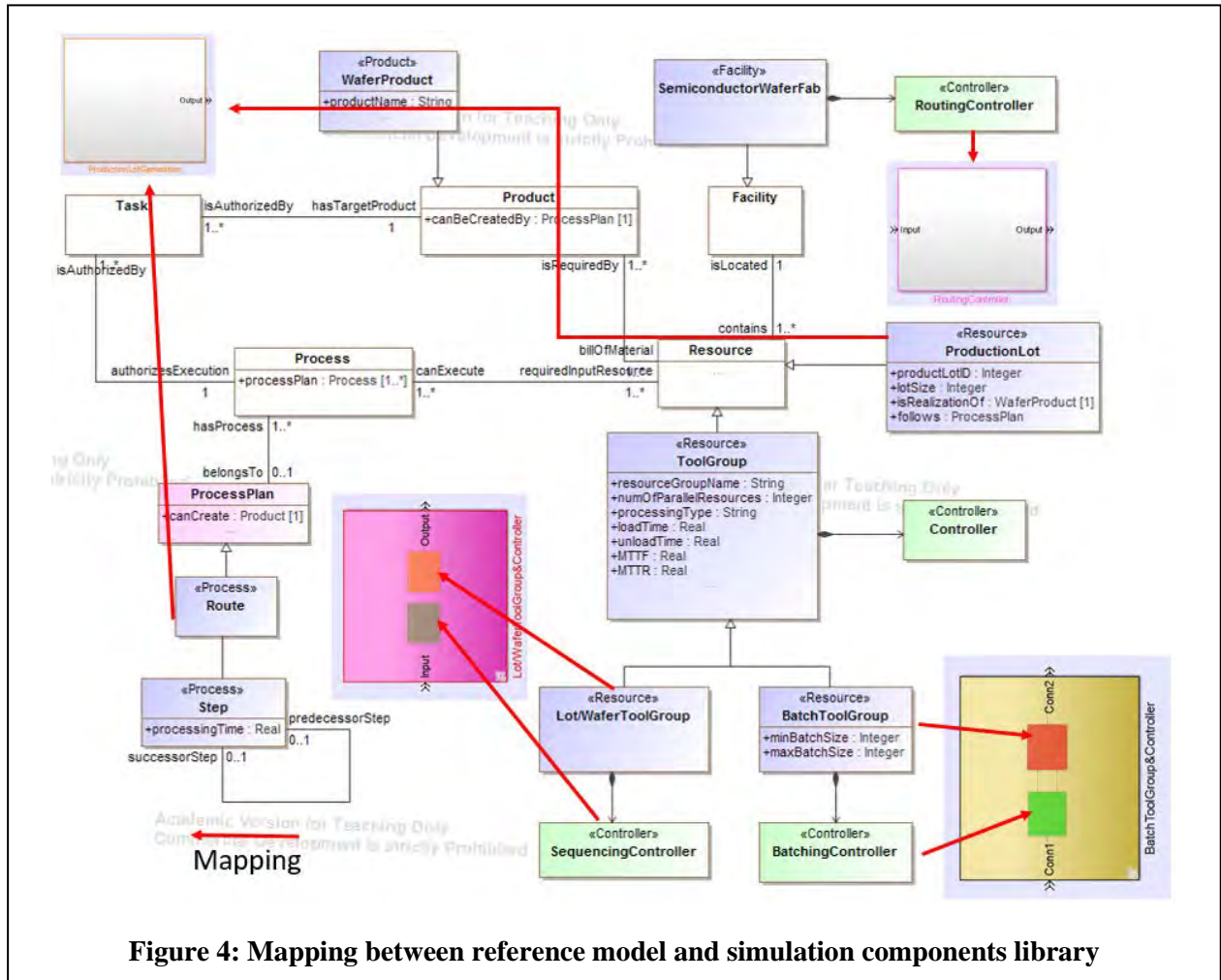
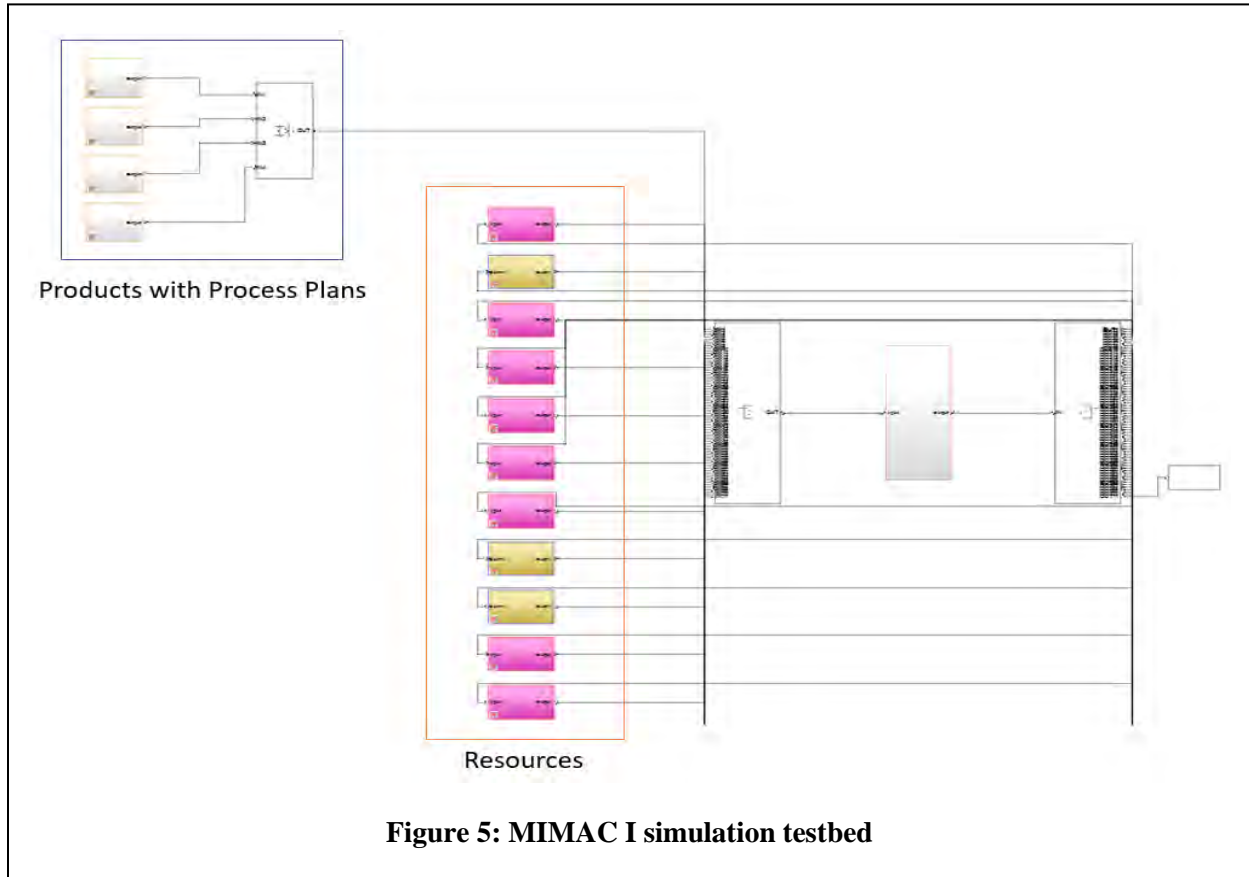


Figure 4: Mapping between reference model and simulation components library

However, generating only one test problem is not very interesting since solving a single problem gives no insight into sensitivity to structure of a test problem. In general, we may want to generate families of test problems to identify, for example what is the best product mix ratio. A simple example is to test how product mix impacts the average overall cycle time of each product. Table 2 shows the results and indicates that products produced less frequently tend to have lower cycle times.





For this simple testbed, we also can easily change the arrival rate of each product over time or have a release schedule for each product that is time bucket oriented. In addition, we also can change the number of parallel identical tools of each tool group. Alternative control strategies also are easily implemented.

In MIMAC I, the process plans are given for each product. If there were more process plans, it would be trivial to increase the number of products in the simulation. In this regard, our approach makes test problems like MIMAC I “scalable” in terms of numbers of process tools, numbers of products, and numbers of jobs in process.

**Table 2: Average cycle time (min)**

<i>Product Mix</i>	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>
<i>2:1:2:2</i>	42427.68	42781.87	50013.65	53288.63
<i>1:2:2:2</i>	32305.18	50552.86	45892.35	47231.56
<i>2:2:1:2</i>	40993.35	49408.25	33769.42	48194.28
<i>2:2:2:1</i>	40109.11	51521.29	40441.57	33724.45

## 6 IMPLICATIONS AND FUTURE WORK

The work reported here is based on the belief that, just as there are standards for representing integrated circuits, there can be standards for representing semiconductor wafer fabs. We have illustrated this for the case of some widely used wafer fab test problems. We have also demonstrated that, given a reference model, data conforming to the reference model, and simulation components conforming to the reference

model, it is relatively straightforward to automate the generation of fab simulation models, at least when there are simple controls.

Our demonstration has been very simple, in part because the standard test problems are very simple. Many extensions are relatively straightforward, such as: adding a randomized time for transport between processes, based on origin and destination, or exploring alternative batching or sequencing strategies which conform to the strategy interface.

Other extensions are not so straightforward. For example, the inclusion of stockers significantly complicates control modeling, because now operational control decisions are, in effect, editing the route for a particular product lot. Yet this is exactly the problem that real fabs must solve. The integration of controllers at different levels of the hierarchy—e.g., fab controllers that see the state of every tool group and individual tool group controllers—also presents significant challenges. Alternative control architectures, such as contract nets, need to be explored. All of these are interesting, but to be explored need to have access to fast, cheap simulation, which automation can help to provide.

Researchers in this domain would significantly benefit from the availability of standard testbeds—not just standard test problems, but a standard “virtual fab” in which alternative proposals for supporting a particular kind of decision could be tested and directly compared. This work demonstrates that such standard testbeds are not out of the realm of possibility.

## 7 ACKNOWLEDGEMENTS

Some of the work reported here was supported by the National Institute of Standards and Technology, under grant 70NANB15H234. In addition, the work has benefited from the availability to software provided by NoMagic, Inc, and MathWorks.

## 8 REFERENCES

- Asmundsson, J., R. L. Rardin, and R. Uzsoy. 2006. “Tractable Nonlinear Production Planning Models for Semiconductor Wafer Fabrication Facilities.” *IEEE Transactions on Semiconductor Manufacturing* 19 (1): 95–111.
- Barua, A., N. Raghavan, A. Upasani, and R. Uzsoy. 2005. “Implementing Global Factory Schedules in the Face of Stochastic Disruptions.” *International Journal of Production Research* 43 (4): 793–818.
- Chen, J., Y. Fan, and C. Chen. 2009. “Capacity Requirements Planning for Twin Fabs of Wafer Fabrication.” *International Journal of Production* 47 (16): 4473–4496.
- Crist, K., and R. Uzsoy. 2011. “Prioritising Production and Engineering Lots in Wafer Fabrication Facilities: A Simulation Study.” *International Journal of Production Research* 49 (11): 3105–3125.
- Dabbas, R. M., H. N. Chen, J. W. Fowler, and D. Shunk. 2001. “A Combined Dispatching Criteria Approach to Scheduling Semiconductor Manufacturing Systems.” *Computers & Industrial Engineering* 39 (3): 307–324.
- Dabbas, R. M., and J. W. Fowler. 2003. “A New Scheduling Approach Using Combined Dispatching Criteria in Wafer Fabs.” *IEEE Transactions on Semiconductor Manufacturing* 16 (3): 501–10.
- Dabbas, R. M., J. W. Fowler, D. A. Rollier, and D. Mccarville. 2003. “Multiple Response Optimization Using Mixture-Designed Experiments and Desirability Functions in Semiconductor Scheduling.” *International Journal of Production Research* 41 (5): 939–961.
- Fowler, J., and J. Robinson. 1995. “Measurement and Improvement of Manufacturing Capacities (MIMAC): Final Report.”
- Fowler, J. W., L. Mönch, and O. Rose. 2006. “Scheduling and Simulation.” In *Handbook of Production Scheduling*, 109–133.
- Jampani, J., and S. J. Mason. 2010. “A Column Generation Heuristic for Complex Job Shop Multiple Orders per Job Scheduling.” *Computers & Industrial Engineering* 58 (1): 108–118.
- Kacar, N. B., D. F. Irdem, and R. Uzsoy. 2012. “An Experimental Comparison of Production Planning

- Using Clearing Functions and Iterative Linear Programming-Simulation Algorithms.” *IEEE Transactions on Semiconductor Manufacturing* 25 (1): 104–117.
- Kacar, N. B., L. Monch, and R. Uzsoy. 2016. “Modeling Cycle Times in Production Planning Models for Wafer Fabrication.” *IEEE Transactions on Semiconductor Manufacturing* 29 (2): 153–167.
- Kacar, N. B., L. Mönch, and R. Uzsoy. 2013. “A Comparison of Production Planning Formulations with Exogenous Cycle Time Estimates Using a Large-Scale Wafer Fab Model.” In *Proceedings of the 2013 Winter Simulation Conference*, 3731–3744.
- Kacar, N. B., and R. Uzsoy. 2015. “Estimating Clearing Functions for Production Resources Using Simulation Optimization.” *IEEE Transactions on Automation Science and Engineering* 12 (2): 539–552.
- Kayton, D., T. Teyner, C. Schwartz, and R. Uzsoy. 1997. “Focusing Maintenance Improvement Efforts in a Wafer Fabrication Facility Operating under the Theory of Constraints.” *Production and Inventory Management Journal* 34 (4): 51–57.
- Kempf, Karl. G. 1994. “Intel Five-Machine Six Step Mini-Fab Description.”
- Mason, S. J., J. W. Fowler, and M. W. Carlyle. 2002. “A Modified Shifting Bottleneck Heuristic for Minimizing Total Weighted Tardiness in Complex Job Shops.” *Journal of Scheduling* 5 (3). John Wiley & Sons, Ltd.: 247–262.
- Mason, S. J., J. W. Fowler, W. M. Carlyle, and D. C. Montgomery. 2005. “Heuristics for Minimizing Total Weighted Tardiness in Complex Job Shops.” *International Journal of Production Research* 43 (10): 1943–1963.
- MathWorks SimEvents. 2015. “MathWorks SimEvents Version 2015b.” *MathWorks*. <https://www.mathworks.com/products/simevents.html>.
- McGinnis, L., and Volkan Ustun. 2009. “A Simple Example of Sysml-Driven Simulation”. In *Winter Simulation Conference*, edited by R. R. H. M. D. Rossetti, B. Johansson, A. Dunkin and R. G. Ingalls,, 1703-1710.
- Mönch, L., and R. Drießel. 2005. “A Distributed Shifting Bottleneck Heuristic for Complex Job Shops.” *Computers & Industrial Engineering* 49 (3): 363–380.
- Mönch, L., J. W. Fowler, and S. Mason. 2013. *Production Planning and Control for Semiconductor Wafer Fabrication Facilities: Modeling, Analysis, and Systems*. Springer Science & Business Media.
- Mönch, L., and I. Habenicht. 2003. “Factory Scheduling and Dispatching: Simulation-Based Assessment of Batching Heuristics in Semiconductor Manufacturing.” In *Proceedings of the 35th Conference on Winter Simulation*, 1338–1345.
- Mönch, L., R. Schabacker, D. Pabst, and J. W. Fowler. 2007. “Genetic Algorithm-Based Subproblem Solution Procedures for a Modified Shifting Bottleneck Heuristic for Complex Job Shops.” *European Journal of Operational Research* 177 (3): 2100–2118.
- Mönch, L., J. Zimmermann, S. J. Mason, and J. W. Fowler. 2011. “Multiple Orders per Job Formation and Release Strategies in Large-Scale Wafer Fabs: A Simulation Study.” *Journal of Simulation* 5 (1): 25–43.
- OMG SysML. 2015. “OMG Systems Modeling Language (OMG SysML) Version 1.4.” *Object Management Group*. <http://www.omg.org/spec/SysML/1.4/>.
- Pfund, M. E., H. Balasubramanian, J. W. Fowler, S. J. Mason, and O. Rose. 2008. “A Multi-Criteria Approach for Scheduling Semiconductor Wafer Fabrication Facilities.” *Journal of Scheduling* 11 (1): 29–47.
- Ponsignon, T., and L. Monch. 2012. “Using Iterative Simulation to Incorporate Load-Dependent Lead Times in Master Planning Heuristics.” In *Proceedings of the 2012 Winter Simulation Conference (WSC)*, 1–12.
- Qi, C., A. I. Sivakumar, and S. B. Gershwin. 2009. “An Efficient New Job Release Control Methodology.” *International Journal of Production Research* 47 (3). Taylor & Francis Group: 703–731.
- Qi, Chao, A.I. Sivakumar, and S.B. Gershwin. 2008. “Impact of Production Control and System Factors in

- Semiconductor Wafer Fabrication.” *IEEE Transactions on Semiconductor Manufacturing* 21 (3): 376–389.
- Scholz-Reiter, B., and J. Heger. 2009. “Analysis and Comparison of Dispatching Rule-Based Scheduling in Dual-Resource Constrained Shop-Floor Scenarios.” *Proceedings of the World Congress on Engineering and Computer Science 2*: 20–22.
- Schönherr, O., and Oliver Rose. 2009. “First Steps Towards a General Sysml Model for Discrete Processes in Production Systems”. In *Winter Simulation Conference*, edited by R. R. H. M. D. Rossetti, B. Johansson, A. Dunkin and R. G. Ingalls,, 1711-1718.
- Sourirajan, K., and R. Uzsoy. 2007. “Hybrid Decomposition Heuristics for Solving Large-Scale Scheduling Problems in Semiconductor Wafer Fabrication.” *Journal of Scheduling* 10 (1): 41–65.
- Sprock, T. 2015. “A Metamodel of Operation Control for Discrete Event Logistics Systems.” Ph. D. Thesis, Georgia Institute of Technology, Atlanta, GA.
- Wein, L. M. 1988. “Scheduling Semiconductor Wafer Fabrication.” *IEEE Transactions on Semiconductor Manufacturing* 1 (3): 115–130.
- Yao, S., Z. Jiang, N. Li, H. Zhang, and N. Geng. 2011. “A Multi-Objective Dynamic Scheduling Approach Using Multiple Attribute Decision Making in Semiconductor Manufacturing.” *International Journal of Production Economics* 130 (1): 125–133.
- Zimmermann, J., and L. Mönch. 2006. “Simulation-Based Selection of Machine Criticality Measures for a Shifting Bottleneck Heuristic.” In *Proceedings of the 38th Conference on Winter Simulation Conference*, 1848–1854.

## AUTHOR BIOGRAPHIES

**PO-CHEN LIN** is a Postdoctoral Fellow in the W. M. Keck Virtual Factory Lab. His e-mail address is [plin72@gatech.edu](mailto:plin72@gatech.edu).

**LEON MCGINNIS** is Professor Emeritus at Georgia Tech, and director of the W. M. Keck Virtual Factory Lab. His research interests are in the merging of mathematics, computing, and modeling to support decision making for the design, planning and control of discrete event logistics systems. His e-mail address is [leon.mcginnis@gatech.edu](mailto:leon.mcginnis@gatech.edu).