

A layered structure for modeling manufacturing processes with the inclusion of energy consumption

Artur Schmidt¹, Thorsten Pawletta¹, Peter Junglas²

¹Hochschule Wismar, Forschungsgruppe CEA

²PHWT Vechta/Diepholz/Oldenburger

{artur.schmidt,thorsten.pawletta}@hs-wismar.de, peter@peter-junglas.de

In the design of manufacturing processes the consideration of resource usage, especially energy consumption, is getting more attention. However, the inclusion of the relevant physical processes in a unified modeling approach is often a non-trivial task. The commonly used simulation tools in this domain usually only support discrete event modeling techniques, while models of physical processes are often continuous in nature and are modeled using ordinary differential equations (ODEs) or differential algebraic equations (DAEs). Moreover, the investigation of such physical processes demands a more specific consideration of process control operations, which are favorably modeled using state machines. To combine the diverse paradigms in a clear-cut way, a layered structure is proposed, which encapsulates the different levels of the model and defines a precise interface between them. The example of an industrial furnace serves as a proof of concept and demonstrates its versatility by integrating a transaction-based material flow model with a state machine describing the process control and physical modeling of the energy flows.

1 Introduction

Models of dynamic systems are based on a simplification of reality. A basic problem is the allowable degree of simplification, which depends on the purpose for which a model is developed. The model must meet the accuracy requirements, but should also be as simple as possible. This means that: (i) the answers delivered by the model have to be true for the current purpose; and (ii) the model should be simple enough that working with it is easy [1].

The modeling and simulation of manufacturing process chains has been a subject of study for several decades. According to [2], the typical modeling approach is discrete event modeling in this domain. This fact is reflected in the popular simulation tools applied in this field today, which particularly support discrete event modeling methods. However, the situation has recently been changing, because of new aspects that have been taken into account and increasing requirements for accuracy.

One of these aspects is the time-dependent energy consumption of single process operations or of total process chains. Approaches for single process opera-

tions, such as in [3], are focused on the energy consumption of single machine operations and they are often based on differential algebraic equations. However, approaches for investigating several machines coupled in a process chain use more abstract models mostly based on discrete event methods, such as in [4] and [5]. Today most approaches related to energy processes in manufacturing are only focused on the simulation of energy consumption. In [6] it is emphasized that the energy consumption of a production line has to be considered in production planning and scheduling. However, the energy consumption then has to be examined together with all the other production performance indicators, such as through-put time, load factors etc.

At this point it is obvious that the decision regarding the necessary degree of simplification for a model takes on a new dimension, because the number of essential variables and evaluation parameters increases. Moreover, it is necessary to determine how finely grained the approximations for continuous energy consumption processes should be. So far, manufacturing models considering production planning problems are mostly discrete event-oriented and the popular simulation environments used in this domain do

not support continuous process modeling.

In [7] a simulator coupling is proposed to execute manufacturing models with mixed discrete event and continuous process behavior. However, the approach is a customized solution and it shows well-known problems of simulator couplings. A hybrid modeling approach based on the Discrete Event and Differential Equation System Specification (DEV&DESS) in [8] is discussed in [9]. It extends the DEV&DESS by an inline integration method that schedules the integration time like discrete events. Thus, continuous processes can be modeled using ordinary differential equations and are solved within a discrete event-oriented simulation environment. However, both approaches are limited according to the modularity and clear separation between model specification and simulation execution. A real modular and accurate problem solution requires a hybrid simulation environment. This paper introduces a novel modular, layered modeling approach for complex hybrid systems implemented in the MATLAB/Simulink [10] environment and using the toolboxes SimEvents [11], Stateflow [12] and Simscape [13]. It supports component-based modeling, exemplified by the example of manufacturing components. Each component consists of a layered structure with sub-components for mapping the internal: (i) material flow; (ii) control operations; and (iii) physical machine operations. Sub-components of a layer can be easily replaced by other sub-components if the interfaces are compatible. Hence, several sub-components with different modeling methods or levels of detail can be developed for a component and alternatively used in a model depending on the goal of the simulation study.

The paper introduces the different layers describing a production component and discusses typical modeling methods for each layer in a general way. Then the approach is illustrated using the application of an industrial furnace. For the industrial furnace a library with several sub-components for each layer is introduced. Finally, some simulation results of various furnace models with different modeling methods and levels of detail are presented.

2 Layers of abstraction for describing production facilities

The *multilayered architecture* is a common design pattern used in software development to cope with the

complexity of the system to be modeled [14]. Each layer represents a different aspect of the system with well-defined interaction between the layers. For the description of a production facility three layers immediately come to mind (Fig. 1):

- The *material flow* layer describes the flow of the parts into and out of the facility. It is the basic layer for connecting facilities to a production line or process chain.
- The *process control* layer maps the local process control of the facility. This is especially important for facilities with complicated manufacturing processes, where the parts are handled in several physical process phases, which may be iterated according to an internal control program or other internal conditions.
- The *process physics* layer implements the physical processes that are relevant in the different manufacturing phases.

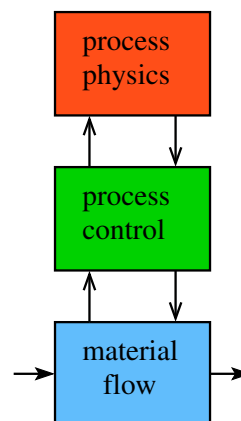


Figure 1: Layers describing a production facility

It is important to adapt the level of abstraction for each layer to the questions at hand and only include the processes that are needed to answer them. Each layer provides its own kind of information: The material flow is mainly concerned with logistic quantities such as waiting times and utilizations; the control layer gives the timing of the different manufacturing phases, which can be useful in the context of other information. The kind of output values of the physics layer can vary widely depending on the implemented details. For the investigation of energy-related questions that are studied and exemplified here, one is mainly interested in the total energy consumption. But

more detailed information can be used too, e.g. the energy used in single internal phases for process optimization or the exact timing of power requirements to take advantage of time varying energy costs.

3 Modeling approaches corresponding to the layers

The layers described above typically utilize very different kinds of abstraction, which demand diverse modeling approaches. A simulation platform for the complete model therefore needs to support a wide range of modeling paradigms.

The material flow layer is often described using the transaction-based modeling method, a special kind of a process-based approach, which defines abstract entities moving between stationary components and acting on them [8]. The entities are identified in manufacturing systems with workpieces or tools, components moved between the production facilities, thus making the material flow apparent.

A convenient method to describe the different manufacturing phases in a facility, which are the concern of the process control layer, are state graphs [15]. The phases directly correspond to the states and the transitions describe the internal process logic. Alternatively, one could again use a process-based approach, wherein the entities now do not denote physical parts, but abstract control tokens. Their flow through stationary components reflects the changes of the internal phases.

The physical manufacturing processes are often modeled as continuous systems, wherein the basic physical laws, e.g. from mechanics or thermodynamics, lead to differential equations (ODEs). If the description contains algebraic constraints or the equations are constructed automatically using a physical modeling approach [16], then the mathematical model is enlarged to a system of differential algebraic equations (DAEs).

For the simulation of the logistic and process-oriented aspects of a production line several discrete simulation programs exist and are in wide industrial use, such as Arena [17] from Rockwell Automation and Plant Simulation [18] from Siemens. Usually these programs lack components such as ODE or DAE solvers to cope with continuous system specifications. This is why different simulators are coupled to solve such problems, such as in [7]. The introduced layered structure

(Fig. 1) can support such simulator couplings, but it cannot solve the general numerical problems of a simulator coupling.

Sometimes a fine-grained physical description of a production facility is already available, e.g. a CAD model inside a finite element program. In such a case the layered structure also supports problem solving using simulator couplings. But this method usually has several severe drawbacks; besides being very cumbersome to handle and leading to very long execution times, the level of physical detail is often too high for the questions in which we are interested here. The extraction of the relevant results from a plethora of computed numbers is often a non-trivial task.

Instead, one should use a system that is capable of hybrid modeling to create a model with the appropriate level of detail. One widely used software is the Matlab/Simulink environment. Basically designed for the simulation of continuous systems using the signal flow paradigm, it can be extended using additional toolboxes and blocksets to include discrete and discrete event modeling features: SimEvents implements components for process-based modeling; Stateflow provides state charts. Both are fully integrated in the Simulink environment. To expand the continuous tool chest one can add physical modeling features with the Simscape blockset and language. These toolboxes/blocksets will be used in the following to validate the layered structure approach by implementing concrete examples.

An alternative choice could be to use a Modelica-based solution [16]. Though mainly designed for continuous modeling with the physical modeling paradigm, it contains basic hybrid features. They have been used to include an additional package for state charts [19] and the package DEVSLib [20] which allows for defining discrete event systems using Zeigler's basic DEVS formalism [8]. Based on this, one can implement a ready-to-use Modelica package similar to Matlab's SimEvents blockset. A corresponding Modelica library is in preparation [21].

4 Application to an industrial furnace

For the investigation of energy efficiency an industrial furnace is an especially rewarding example; it is widely used in metalworking processes for hardening,

annealing and tempering and is often one of the most extensive energy consumers. In addition, its internal operation is generally rather complex, making it an ideal example for the application of different modeling approaches. The concrete operation of the furnace is patterned in the following after the descriptions in [9] and [22].

When parts enter the furnace, they are collected until a given batch size is reached. A complete batch then enters the oven, is subjected to possibly several phases of heat treatment and leaves the furnace. This completes the description of the material flow.

The process control layer should map six phases (Fig. 2).

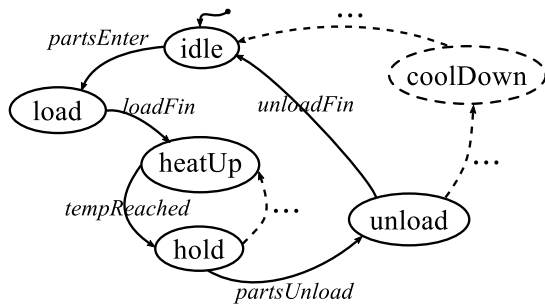


Figure 2: State graph describing the internal furnace processes

The *idle phase* spans the times before parts have entered and after all parts have left the oven. During the *load phase* parts enter the oven. In the *heat-up phase* the oven is heated until a given temperature is reached; this is then held constant during the *hold phase*. According to the requirements, the heat-up and hold phases can be iterated several times with different temperatures. Finally, the parts leave the oven in the *unload phase*. An additional *cool-down phase* may be included either to make sure that the parts leave the oven with a moderate temperature or to describe a shutdown of the oven.

Since the focus of this investigation lies on the energy consumption, the process physics layer is concerned with the modeling of the heat flows in the furnace. The only source is the power supply of the actual heater. From here the heat flows mainly through convection and radiation processes to the parts and to the internal structures and the casing of the oven. During the heat-up and hold phases losses are mainly due to conduction through the casing into the environment, while in the load or unload phases additional losses are caused

by the open doors. Because of the complicated geometry, the physical details, especially of the convection processes, are also rather complicated. However, for the estimation of the total heat flows common approximate methods usually give quite accurate results.

5 Designing a library for the furnace

To show how to realize the abstract ideas in practice a set of layered models will be described using modeling methods supported by the Matlab/Simulink environment. The inclusion of a complete furnace component in a production line or process chain is illustrated in Figure 3. The production line is reduced for simplicity to a minimal structure composed of: (i) a source component for generating the parts; (ii) a queue component in front of the furnace; and (iii) a sink component for handled parts. On this level the system is implemented as a discrete event model with SimEvents using its entity flow approach to describe the flow of parts into and out of the furnace.

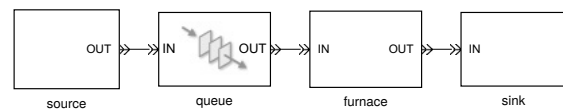


Figure 3: Simple discrete event-based production line model with a furnace component in SimEvents

Figure 4 shows the internal composition of the furnace component according to the introduced layered structure. It consists of the three subsystems: (i) MF for the material flow; (ii) PC for the process control; and (iii) PP for the process physics. The subsystems have to communicate in several ways:

- MF receives parts from the external input port and sends to the PC the number of parts that have entered the MF. When the batch size is reached the PC starts the physical process (PP) which models the different manufacturing phases during the operation of the furnace.
- When the last manufacturing phase (“unload”) is finished, the PC sends a “leaving” signal to the MF, which accordingly forwards the processed parts to its external output port.
- During the operation of the furnace the PC signals the current manufacturing phase to the PP,

which adapts the internal heat flows accordingly. This can mean changing the supplied heat between the heating and holding phases or increasing the losses due to the doors being open while loading or unloading.

- In return the PP sends the current temperature values of the furnace and the parts to the PC, which uses them to determine whether the heat-up phase or the optional final cool-down phase is complete.

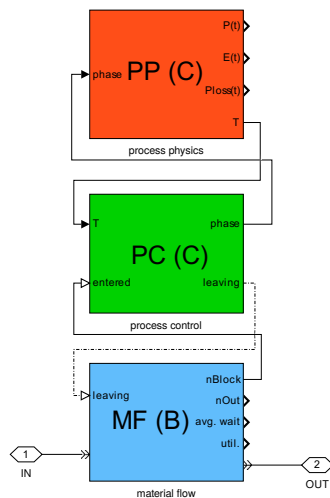


Figure 4: Layered structure of the furnace component

Each layer has its specific set of parameters and output quantities: The MF subsystem defines the batch size and logistic properties such as average waiting time and machine utilization; the PC subsystem gets the heating program and outputs the time in the different phases. The PP system needs a lot of physical parameters for the calculation of the heat flows and provides the power requirements during the process as well as the total energy consumption.

To prove the concept and provide some concrete examples a basic library has been implemented (Fig. 5), containing several subsystems for each layer, which use different modeling methods or have varying levels of detail. They are labeled with the two letters denoting the layer and a third in brackets giving the complexity in ascending order, with (A) being the simplest model:

- The basic material flow subsystem MF(A) uses only a simple server, while MF(B) explicitly contains an input tray, where the batch is compiled.

- The simplest process control subsystem PC(A) uses a simple SimEvent model to describe one pass through the four basic phases (ignoring idle and cool-down states). PC(B) adds a loop to allow a repetition of heat-up and hold phases according to its heat program parameter, implemented as a SimEvents model. Additionally, PC(C) adds a cool-down phase, but because of the more complex control logic it is implemented using the state machine approach with Stateflow.
- The process physics subsystem PP(A) uses only the internal oven temperature and a simple formula for the global losses, while PP(B) adds the temperature of the parts, the heat transfer between oven and parts and additional losses during the load and unload phases. Both use standard Simulink methods to implement the corresponding differential equations. Finally, PP(C) employs physical modeling which is modeled using Simscape for the same physical processes as PP(B). This makes the physical model structure more transparent and easier for engineers to expand.

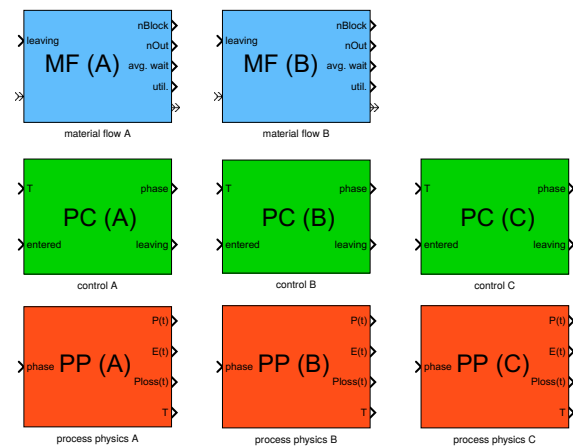


Figure 5: Library with subsystems to compose different variants of a furnace component

6 Implementation details

In the following, three different variants of a furnace component will be described in more detail: (i) the very basic model ovenBAA – the letters stand for the composition of MF(B), PC(A) and PP(A) components; (ii) a medium complex model ovenBCA; and

(iii) the most complex variant ovenBCC. While this section is devoted to the implementation of the components using the MATLAB/Simulink tools, the next one will discuss some simulation results.

6.1 Material flow

All of the model examples considered here use the subsystem MF(B) for mapping the internal material flow. The model structure of subsystem MF(B) (Fig. 6) consists of two simple servers: the first for the input tray; the second (N-Server) for the oven proper. Both hold incoming entities (i.e. parts) up to the given batch size, until their succeeding gates open to transfer the entities to the next stage. The intermediate Gate guarantees that a full batch is always delivered to the oven; the final Release Gate is triggered by the process control layer at the end of the unload phase.

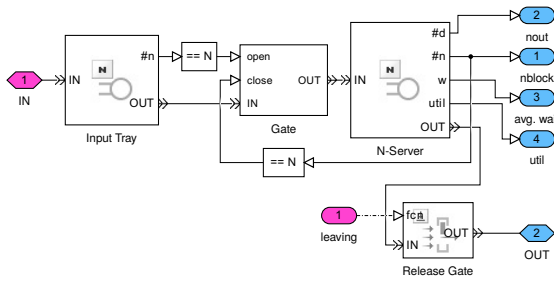


Figure 6: Discrete event-based model of MF(B) using SimEvents

6.2 Process control

The process control subsystem starts when the number of parts that have entered at the material flow layer is equal to the batch size. In subsystem PC(A) (Fig. 7), implemented using SimEvents, this leads to the creation of a control entity that passes through a line of servers denoting the different phases. Except for one, all servers simply have a fixed processing time; only the heat-up phase is different. Here the entity is held in the server until the current oven temperature, which is computed in the upper physics layer, has reached the given temperature T_{set} . After the unload phase, the control entity is destroyed and the wake-up signal (trigger) is created, which opens the Release Gate at the material flow layer.

The much more elaborate subsystem PC(C) (Fig. 8)

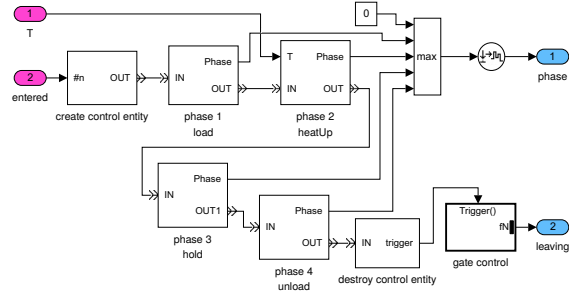


Figure 7: Discrete event-based model of PC(A) using SimEvents

incorporates all six phases, pictured in Figure 2, as well as possible replications of the heat-up and hold phases. It is a hybrid DEV&DESS model based on the state machine approach, but including continuous state event handling. It is mainly implemented using Stateflow. The two auxiliary subsystems create events when the waiting time has changed or the heat-up or cool-down temperature has been reached. The model PC(C) is essentially an adapted version of a model described in [9].

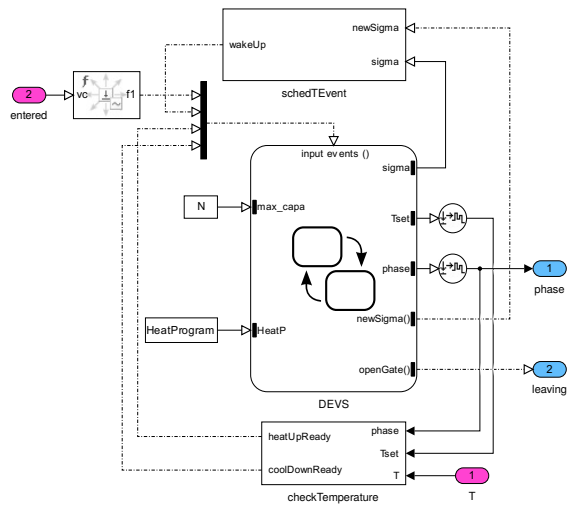


Figure 8: Discrete event-based model with continuous state event handling of PC(C) mainly using Stateflow

6.3 Process physics

The basic process physics subsystem PP(A) (Fig. 9) uses a simple power balance to compute the change of the oven temperature T_o :

$$C_o \dot{T}_o = P_{heat} - P_{loss},$$

where C_o is the total heat capacity of the oven. The power loss is computed with Newton's simple law of cooling:

$$P_{loss} = k_a(T_o - T_e),$$

where T_e is the temperature of the surroundings and k_a a constant that subsumes all convective and conductive processes. The heating power is assumed to be constant during the heat-up phase and to match the losses during the hold phase:

$$P_{heat} = \begin{cases} P_H & \text{heat-up} \\ k_A(T_o - T_e) & \text{hold} \end{cases}$$

For the computation of the total power demand of the furnace, a constant power P_0 is added, which subsumes all non-heating processes, such as a base load or the power electronics.

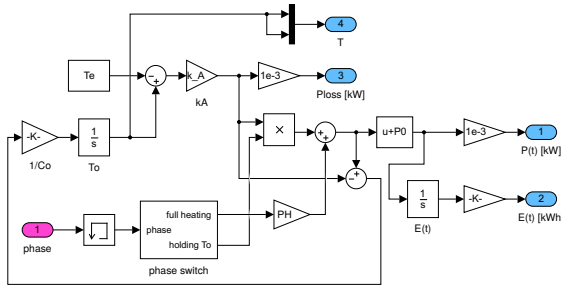


Figure 9: ODE-based model of PP(A) using Simulink

The subsystem PP(C) (Fig. 10) uses physical modeling based on Simscape to incorporate much more physical details, such as convection and radiation from the oven to the load and – during load and unload phases – to the environment. The governing differential equations are not built up explicitly here; instead, the physical components such as heat capacities and various kinds of heat flow are represented directly, which makes the physical structure of the model much clearer.

6.4 Implementation pitfalls

The extension of Simulink by several packages is necessary to make a multi-paradigm model possible, but it also leads to small inconsistencies, which have to be overcome. A minor nuisance here is the large number of different ways used to express values and signals in the packages used: Simple scalar values can be time-based in Simulink, event-based in SimEvents or physical-valued in Simscape. An event can be defined

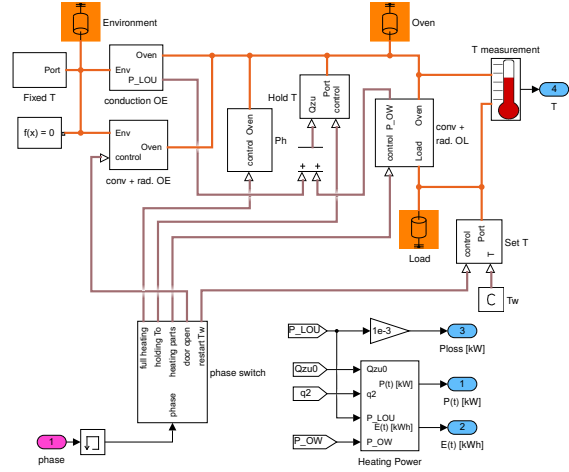


Figure 10: Physical model (DAE) of PP(C) using Simscape

by a sample time hit, a rapid value change (“edge”), a special trigger signal or a function call.

To clearly define a common interface for the layer components, their external values are fixed in the following way: The temperature and phase values are time-based; the number of parts is event-based; and the “leaving” signal is a function call. If the internal implementation of a block generates differing types, then one has to use one of the many converter blocks to get the proper kind of signal, e.g. the round Event to Timed Signal blocks that can be seen in Figures 7 and 8.

This problem is especially annoying in the physical modeling environment. The large number of necessary converters and reference points clutters the model and destroys the clear physical structure. Hiding them in subsystems is an obvious way to regain an ordered visible representation of the underlying physics model.

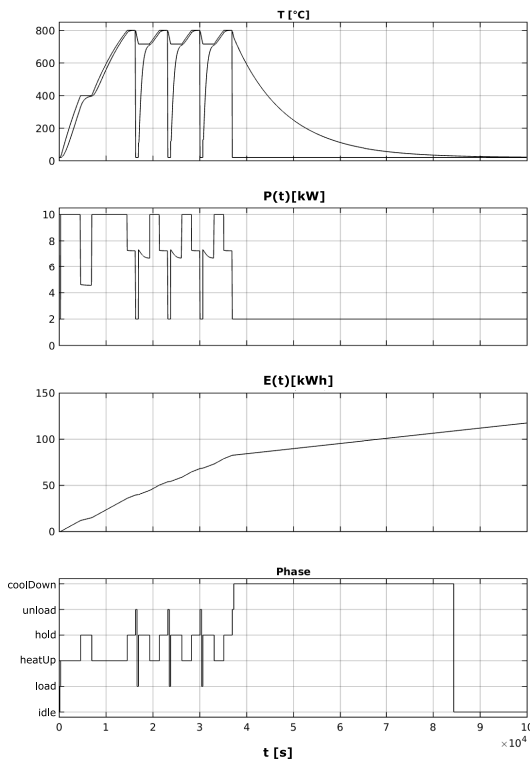
7 Simulation results

In a simple example experiment 24 parts were processed using a batch size of six. After the heat-up phase their temperatures were held first to 400 °C for 40 minutes, then to 800 °C for 30 minutes. Figure 11 shows some simulation results of the model variant ovenBCC: the temperatures of the furnace and parts, the power consumption of the furnace, the accumulated used energy and the current internal manufacturing phase, each as functions over time.

The results are similar to those presented in [9], but ovenBCC incorporates more details, especially of the physical model layer. The other two example models ovenBAA and ovenBCA basically reproduce the results from [9], since they are based on the same physical model assumptions. For comparison, Table 1 shows the total energy use E_1 at the exit of the last part and E_2 at the end of the simulation; Figure 12 displays the power consumption for the three models.

Model	E_1 [kWh]	E_2 [kWh]
ovenBAA	71.3	108.8
ovenBCA	69.0	106.6
ovenBCC	82.6	117.6

Table 1: Comparison of the energy consumption



Time offset: 0

Figure 11: Simulation results of model variant ovenBCC

The higher energy needs in variant ovenBCC are due to the heating of the parts, which enter the oven with the low temperature of the environment – an effect that has not been taken into account in the other models. The small difference between the two simpler model variants results from the different timing of the phases,

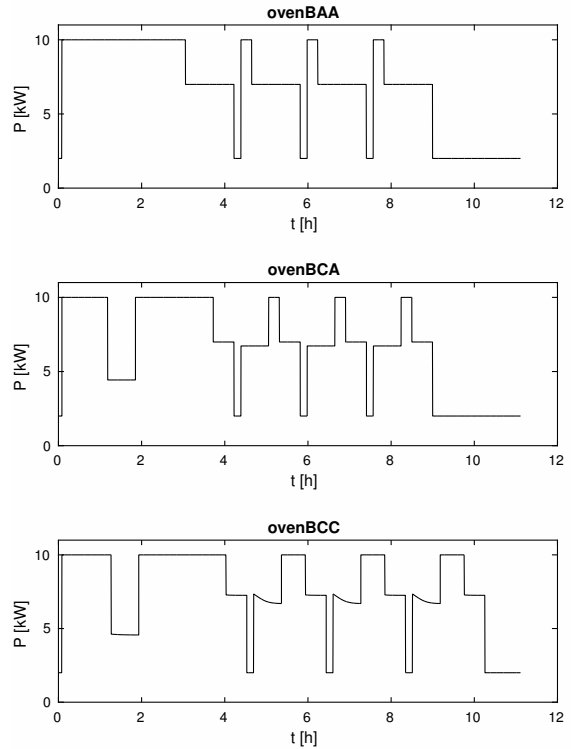


Figure 12: Comparison of the power results

as can be seen from Figure 12. It is interesting to note that the total energy results only differ by 10% - 15%. If this level of accuracy is sufficient for the question at hand, e.g. for a global assessment of a complex production line, then one can use one of the simple physics models. However, Figure 12 shows that for a detailed examination of the power needs during the individual phases one has to stick to the complexities of model variant ovenBCC.

How much do we pay for the additional level of detail in the complex model? For a comparison of run times, the number of parts was raised to 600 and three runs were performed for each model. To get rid of loading and compile times, the means of only the last two results were taken, leading to the values in Table 2.

The results show that the complex physics layer using Simscape costs a factor of two in execution time, which can be reduced to a factor of 1.6 by replacing the physical modeling with standard ODE methods using Simulink (ovenBCB). The small value for ovenBCA comes as a surprise, since its process layer is definitely more complex than that of ovenBAA. The solution to this puzzle is given in the last line of the table: The subsystems PC(B) and PC(C) have nearly

Model	Run time [s]
ovenBAA	25.1
ovenBCA	20.7
ovenBCC	49.8
ovenBCB	39.3
ovenBBA	27.7

Table 2: Comparison of run times

the same complexity, but PC(C) uses Stateflow instead of SimEvents. Apparently, the implementation of SimEvents has some potential for optimization.

8 Conclusion

The introduced layered model structure supports the modular modeling of complex components, such as production facilities, using different methods for specifying the dynamic behavior. Thereby, it provides a natural approach for implementing compatible models with different levels of detail.

Using the example of a model library for an industrial furnace it has been shown that model components of production facilities can be usefully structured in three layers. The first layer maps the internal material flow. It delivers an external input and output interface for connecting with other facilities in a manufacturing process chain model and an internal interface for communication with the second layer. The dynamic of this layer is described as transaction-oriented, a special kind of entity-based discrete event modeling. In the second layer local process control operations are modeled. These operations have mainly discrete or event-based dynamics and are usefully described using transaction or state-based methods. However, depending on the specific process operations, modeled in the third layer, it could be necessary to detect and handle state events of continuous values. Additionally, the process control layer provides event-based control inputs for the first and third layers. The dynamic behaviors of specific process operations can vary significantly and particularly depend on the level of detail. In the case of the industrial furnace the process operations have been described by (i) simple ordinary differential equations and (ii) complex differential algebraic equations using physical modeling methods.

Using the model library, various models for the indus-

trial furnace using different modeling techniques have been implemented and compared relating to their simulation results and runtimes. Until now the experiments have been focused on the furnace as a single production facility and its energy consumption. In the next step the furnace component should be examined in the more complex context of process chains and then further performance indicators should be considered.

Acknowledgements

A. S. and Th. P. express their sincere thanks to the German Research Foundation DFG for funding the research project no. PA 631/2-2

P.J. is grateful for the hospitality extended to him by Tom Schramm and his colleagues at the Department of Geomatics, HCU Hamburg.

References

- [1] D.P.F. Möller. *Mathematical and Computational Modeling and Simulation – Fundamentals and Case Studies*. Springer Pub., 2004.
- [2] E. Lefeber, J.E. Rooda. *Modeling and Analysis of Manufacturing Systems*. In: Handbook of Dynamic System Modeling, P.A. Fishwick, CRC Press, p. 34_1-34_20, 2007.
- [3] C. Eisele. *Simulation-based optimization of the electrical consumption of metal-cutting machine tools (in German)*. Dissertation Thesis, TU Darmstadt, Germany, 2014.
- [4] N. Weinert. *An approach for the planning and operation of energy-efficient production systems (in German)*. Dissertation Thesis, TU Berlin, Germany, 2010.
- [5] R. Larek. *Resource-efficient design of manufacturing process chains by simulation and numerical optimization (in German)*. Dissertation Thesis, Univ. Bremen, Germany, 2012.
- [6] S. Schrems. *A method for the model-based integration of machine-specific energy demand in production planning (in German)*. Dissertation Thesis, TU Darmstadt, Germany, 2014.

- [7] T. Peter, S. Wenzel. *Simulation-based Planning and Evaluation of Energy Efficiency for Production Systems in Car Manufacturing (in German)*. Simulation in Production and Logistics, Fraunhofer Pub., Stuttgart, Germany, 535-544, 2015.
- [8] B. Zeigler, T. G. Kim, H. Praehoffer. *Theory of Modelling and Simulation*. Academic Press, San Diego, 2nd ed. 2000.
- [9] A. Schmidt, T. Pawletta. *Hybride Modellierung fertigungstechnischer Prozessketten mit Energieaspekten in einer ereignisorientierten Simulationsumgebung*. Proc. ASIM 2014 - 22. Symposium Simulationstechnik, Berlin, p. 109-116, 2014.
- [10] The MathWorks. *Simulink: Simulation and Model-Based Design*. Online: www.mathworks.de/products/simulink/ (called 2015-11-11).
- [11] The MathWorks. *SimEvents: Model and simulate discrete-event systems*. Online: www.mathworks.de/products/simevents/ (called 2015-11-11).
- [12] The MathWorks. *Stateflow: Model and simulate decision logic using state machines and flow charts*. Online: www.mathworks.de/products/stateflow/ (called 2015-11-11).
- [13] The MathWorks. *Simscape: Model and simulate multidomain physical systems*. Online: www.mathworks.de/products/simscape/ (called 2015-11-11).
- [14] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal. *Pattern-Oriented Software Architecture, Vol. 1: A System of Patterns*. John Wiley & Sons, New York, 1996.
- [15] D. Harel. *Statecharts: A Visual Formalism for Complex Systems*. Science of Computer Programming, 8, pp. 231-274, 1987.
- [16] P. A. Fritzson. *Principles of Object-Oriented Modeling and Simulation with Modelica 3.3*. Wiley & Sons, New York, 2015.
- [17] W. D. Kelton, R. Sadowski, N. Zupick. *Simulation with Arena*. McGraw-Hill, New York, 6. ed. 2014.
- [18] S. Bangsow. *Tecnomatix Plant Simulation: Modeling and Programming by Means of Examples*. Springer, Berlin, 2016.
- [19] H. Elmqvist, F. Gaucher, S. E. Mattsson, F. Dupont. *State Machines in Modelica*. Proc. 9th Int. Modelica Conference, Munich, p. 37-46, 2012.
- [20] V. Sanz, A. Urquia, F. E. Cellier, S. Dormido. *System modeling using the Parallel DEVS formalism and the Modelica language*. Simulation Modelling Practice and Theory, 18, 998-1018, 2010.
- [21] P. Junglas. *A Modelica library for transaction based modeling*. Work in progress.
- [22] V. Heuer, K. Löser. *Energetische Optimierung von thermochemischen Vakuumprozessen und Anlagen in der Großserie*. Elektrowärme International, 69, pp. 261-268, 2011.