GENERATION OF ALTERNATIVES FOR MODEL PREDICTIVE CONTROL IN MANUFACTURING

Sören Stelzer[(a)], Sören Bergmann[(b)], Steffen Straßburger[(c)]

Ilmenau University of Technology
Helmholtzplatz 3
D-98684 Ilmenau, GERMANY

[(a)]soeren.stelzer@tu-ilmenau.de, [(b)] soeren.bergmann@tu-ilmenau.de, [(c)] steffen.strassburger@tu-ilmenau.de

ABSTRACT

Manufacturing systems are dynamic systems which are influenced by various disturbances or frequently changing customer requests. A continuous process of decision making is required. Model Predictive Control is a common model-based approach for control but needs adaption to be applicable to discrete-event simulation. In this paper we introduce an approach to model and generate non trivial control options and decisions often made in the operation of manufacturing systems. We also show how complex scenarios can be generated. To support a wide-range of applications our approach is based on the core manufacturing simulation data (CMSD) information model. We implement the design and generation of complex scenarios by processing and combining modeled control options. By using our approach, which also applicable to decision support systems, we can enable model-based closed-loop control based on a symbiotic simulation system and automated model generation and initialization.

Keywords: simulation, CMSD-IM, design of experiments, decision support system, model predictive control

1. INTRODUCTION

Manufacturing systems are dynamic systems and subject to various internal and external disturbances, which often influence the expected behavior in an undesired way. Additionally, they have to deal with growing uncertainties, flexibility, and high cost pressure.

These facts lead to changing circumstances for the decision making process. Decisions have to be made in higher frequency, which directly leads to a shorter time available for finding them. Therefore, a continuous process of decision making and controlling is required to make sure the aimed goals can be achieved. Additionally, the complexity of internal and external processes is rising. For the same reason the amount of gained data in the connected information systems is also increasing.

To keep up with the tightened situation decision makers are forced to use Decision Support Systems (DSS). To also handle the complexity of today’s manufacturing systems discrete event simulation (DES) is used in conjunction with DSS. This leads to model-driven DSS (Heilala et al. 2010). Related approaches are Online-Simulation (Davis 1999; Hanisch, Tolujew, and Schulze 2005), Simulation based Early Warning Systems (SEWS) (Hotz 2007), and Symbiotic Simulation Systems (Aydt et al. 2008a).

In our research work we investigate an automated control approach called Model Predictive Control (MPC) to enable a closed-loop control for manufacturing systems, using techniques like Symbiotic Simulation and SEWS. While MPC is well studied in the field of automatic control engineering, there is very limited research and application of MPC using discrete event simulation techniques. To enable MPC to manufacturing control or decision support, we identified three major research tasks: the formal description of alternatives, the generation of complex scenarios based on combinations of these alternatives, and the generation and appropriate initialization of simulation models with the state of the real system.

In this paper we focus on the methodology of how common decision and control alternatives in the operation of manufacturing systems can be formally described and automatically generated. We also show how based on this description complex scenarios can be designed and generated. Our modeling approach is based on an applied information model, provided by the Core Manufacturing Simulation Data Standard (SISO 2010). Initialization, generation and execution of simulation models have been discussed in previous work (Bergmann, Stelzer, and Strassburger 2011).

The remainder of this paper is structured as follows. In section 2, we discuss related work. In section 3, we illustrate requirements for model-predictive control in manufacturing using discrete event simulation. We also introduce the information model used in our approach. In section 4, we discuss typical decisions and control options used to influence manufacturing systems and how they can be described using the CMSD-IM. In section 5, we describe our methodology of how we intelligently and automatically generate simulation scenarios from the modeled control options. In section 6, we summarize the results of our approach and give an overview about current and future work.

2. DISCUSSION

2.1. Related Work

The goal of decision makers and controllers is to find an optimal solution for a given situation from a set of alternatives. Therefore it is required to determine the current situation as correct as possible and to estimate the impact of all available decisions or control inputs. Due the high complexity of the internal processes and large amount of data, the estimation of the impact of an alternative to the manufacturing system can not be handled anymore by humans (Heilala et al. 2010).

Discrete Event Simulation (Banks et al. 2000) is a well-accepted technique for planning, investigation and operation of manufacturing systems and supply chains (VDI 3633-1). The application of modeling and simulation for manufacturing systems is not a novel approach, but a dominating part of simulation applications are focused on the planning of new systems or their modification. When the actual operating of manufacturing systems is investigated, the discussed applications are often limited to scheduling problems.

To extend the benefits of modeling and simulation to the operation of manufacturing systems a closer integration of simulation techniques and manufacturing systems is needed. In 2004, Fowler and Rose discussed the future of modeling and simulation and defined a couple of challenges for research and applications. Among them, the closer integration and interaction of simulation techniques and information systems was identified as grand challenge (Fowler and Rose 2004).

The closer integration of simulation techniques and manufacturing enables the handling of complex processes and huge amounts of data required for decision support or control. The Online-Simulation approach (Hanisch, Tolujew, and Schulze 2005) focuses on a closer integration of simulation and manufacturing systems by obtaining or keeping simulation models up-to-date with the investigated system. Besides the introduction of different methods for initializing the simulation with data from the real system, Hanisch et al. also discussed requirements and aspects of data and model quality.

In previous work, we have already presented a solution for model generation and initialization as a potential way to obtain up-to-date simulation models (Bergmann and Strassburger 2010; Bergmann, Stelzer, and Strassburger 2011). The chosen approach is based on the Core Manufacturing Simulation Data Information Model (CMSD-IM).

Symbiotic Simulation (Aydt et al. 2008a) and simulation based early warning systems (SEWS) (Hotz 2007) are further approaches which are focusing on the application of a closer integration of manufacturing information systems and simulation environments. While SEWS are an application of a close integration of information systems to monitor manufacturing systems, symbiotic simulation is discussing the possible interactions and benefits of simulation environments and manufacturing systems. Further, the aspects of symbiotic simulation systems enable a wide range of new applications of modeling and simulation.

The goal of both approaches is to enhance the quality of the decision making process. A key benefit of these approaches is the possibility to consider the current state of the system under investigation. This enables situation-based decision support and control, like prediction of the trajectory of crucial processes. It is also possible to predict the behavior of the investigated system after applying a control input or decision. This is also found in literature as “what-if”-analysis (WIA). Aydt et al. (2008b), for example, showed in a semi-conductor manufacturing application how decision support can be enabled by variation of simple model parameters. There is no information given on modeling requirements or how to implement a symbiotic simulation.

The observed applications of simulation-based decision support in manufacturing are limited to the variation of parameters or scheduling. This is primarily caused by the lack of appropriate methods for describing and modeling of complex control options and decisions. Beyond the simple variation of parameters or schedules, there is a wide variety of decisions and control options (discussed in section 4) which can not be described by simple parameters. The definition of complex scenarios often leads to an extensive manual modeling process. This makes it difficult to automatically generate such scenarios and iterate through them. We consider this an important requirement for simulation-based control.

Neglecting these more complex control options, manufacturing control is already applied and discussed in several papers. A closed control loop involving manufacturing execution systems (MES) as controller is suggested by Kletti (2007) (Figure 1). His goal is to use a model of the system to evaluate a set of possible alternatives and choose the optimal one, regarding current objectives. These alternatives should consider the current system state and include possible decisions, strategies or control values. In reality there are no MES applications which implement an automatic control loop using discrete simulation techniques. Instead they used deterministic forward calculations (often wrongfully named “simulation”) and neglect the dynamic and stochastic behavior of the system.

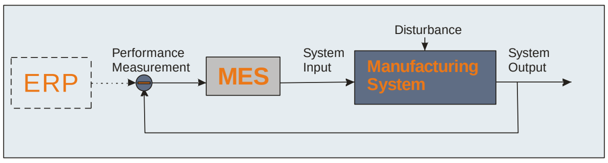

Figure 1: Propagated control loop for manufacturing execution systems (according to Kletti 2007)

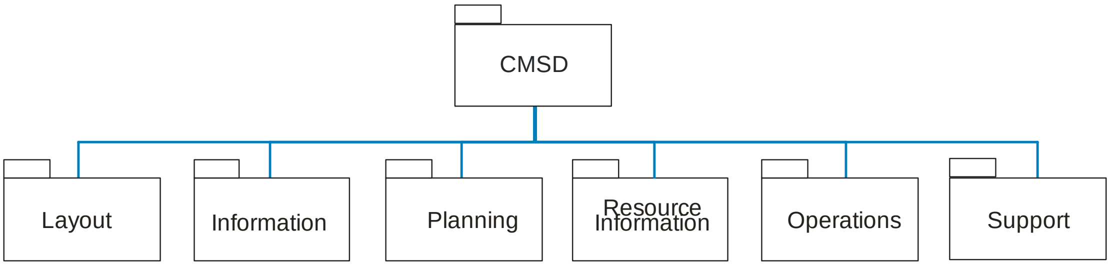

Figure 2: Overview of packages provided by CMSD-IM

There is also related work in the area of automatic control engineering. Finding (semi)-optimal control vectors, analog to alternatives, using a model of the system is called model-predictive control (MPC). Model predictive control was applied to manufacturing systems by Wang et al. They were using analytical models (Wang, Rivera, and Kempf 2007), which are not available and applicable in most cases of manufacturing control and decision support.

To make MPC an accepted approach for manufacturing control it is necessary to use the discrete event simulation technique as a model base. This is caused by the lack of other modeling approaches which can describe the complex dynamic and stochastic properties of systems in this domain.

Regarding the requirements for MPC, symbiotic simulation systems are well suited for establishing a closed-loop control. What is missing is a methodology of how common decision and control alternatives in the operation of manufacturing systems can be formally described and automatically generated.

For this it is necessary to describe and model given alternatives in a way, which allows automated and enumerable combination. With this the controller will be enabled to directly evaluate the search space for finding an optimal scenario.

We also show how based on this description complex scenarios can be designed, generated and afterwards be evaluated through simulation.

Before we describe the modeling of alternatives in CMSD-IM, we have to discuss requirements for model predictive control based on a standard focusing on interoperability of manufacturing information systems and simulation environments.

3. REQUIREMENTS FOR MODEL PREDICTIVE CONTROL

The requirements for enabling a closed-loop control or decision support of manufacturing systems can be separated into four major aspects. Most of them are based on requirements for data exchange, model generation, symbiotic simulation, and online simulation. At first we have to discuss the representation of the investigated manufacturing system and secondly the automated generation and initialization of simulation models. The third aspect is the aggregation of results and the last aspect concerns how to generate complex scenarios to formulate WIAs.

Based on our work on automated model generation and initialization we are using the CMSD-IM as standard for data exchange and modeling manufacturing systems.

3.1. CMSD

The primary objective of the CMSD Information Model is to facilitate interoperability between simulation systems and other manufacturing applications. The CMSD standard provides data structures and an information model (Figure 2) which was designed to firstly support the exchange of modeling information. To cover the complexity of production and logistic systems and a wide range of modeling approaches, the standard allows aspects of the system to be mapped in CMSD in multiple ways.

The capabilities of CMSD were demonstrated in several research projects (Leong et al. 2006; Johansson et al. 2007), which mostly focus on the developing of new and modified production systems. Our own work has focused on using CMSD to support the operational phase of manufacturing systems (Bergmann, Stelzer, and Strassburger 2011).

3.2. Modeling of Manufacturing Systems

The basic idea of model-driven approaches like symbiotic simulation or model predictive control is to use a model of the investigated system to obtain information about its behavior. To assure the correctness of these results, a verified and validated model is needed. In our work we are using the CMSD-IM to store and structure information. This is done by analyzing the real system or collecting information from information systems connected to the real system. Information about processes, resources and materials can, for instance, be imported from Enterprise Resource Planning (ERP) or Manufacturing Execution Systems (MES).

CMSD-IM enables a data-driven modeling approach decoupled from modeling or simulation tools. The information model of CMSD-IM consists of several classes representing common objects found in manufacturing systems like machines and workers. It is also capable of defining process plans as well as representing job and order information.

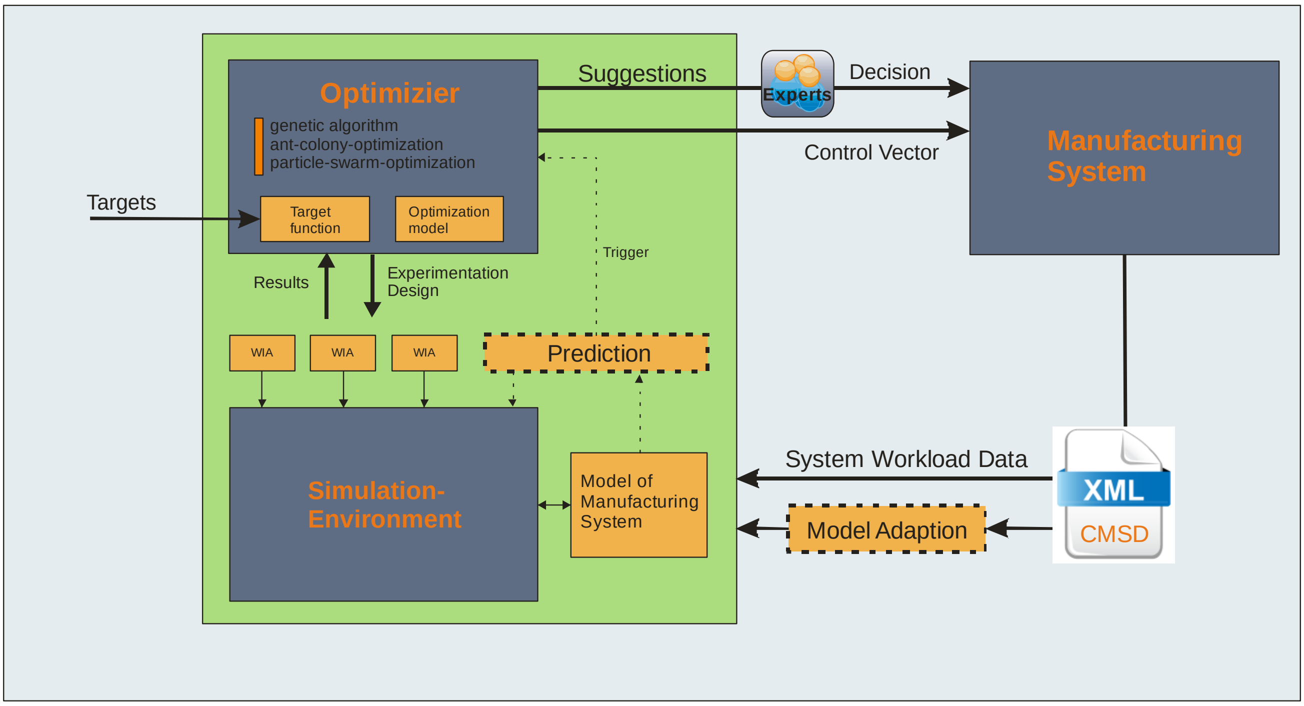

Figure 3: Model Predictive Control / Decision Support System based on automated model generation and initialization

CMSD-IM further allows the checking for logical correctness and therefore facilitates the detection of any missing information extracted from the external information systems.

By using our web-based interface, decision makers can interactively complete necessary data. It is also possible to manipulate the conceptual model via the web-based user-interface.

All objects and attributes of the CMSD-IM are presentable in an XML description using an associated schema. This characteristic enables the automated processing which is required for model generation.

3.3. Description of Alternatives and Generation of complex Scenarios

In contrast to systems which are analytically describable manufacturing systems are modeled using discrete event simulation models. With this, we have no simple way to determine an optimal solution.

In the case of non-trivial manufacturing scenarios, we therefore have multiple dependencies of the system on potentially many parameters. We therefore cannot simply single out a simple variable for optimization.

We therefore require a way to explicitly describe and model the different alternatives, which can afterwards be selected and combined by the controller or the decision support system. To achieve the automated processing of model descriptions we choose CMSD-IM to describe the alternatives (see section 4).

To combine the modeled alternatives it is required to process them and build logically correct and valid scenarios, which afterwards will be automatically generated and executed (section 5).

3.4. Automated Model Generation and Initialization

To support decision makers or enable control to systems using discrete event simulation different opportunities have to be evaluated by simulating them. Therefore an executable simulation model is required. Finding (semi)-optimal solutions to an actual situation typically requires a huge amount of simulation runs, with different model variations.

From previous work we already know that CMSD-IM is well suited for representing manufacturing systems. To transform the CMSD-IM description into executable simulation models we are using an automated model generation approach. The model generation allows the creation of simulation models for different simulation tools, for example Plant Simulation (Siemens 2012) or SLX (Henriksen 1999).

To use simulation as an operational decision support tool we also need to appropriately initialize the simulation model, i.e., a mechanism to keep the simulation model up-to-date with the real system. In previous work we described a methodology based on CMSD-IM to do this. This approach combines the automated model generation and initialization. This is enabled by CMSD-IMs ability to keep system load and state information of modeled entities as well.

3.5. Distribution of Experiments and Result Aggregation

After the scenario is designed and an executable model is built by the automatic model generation, the model has to be executed. In our framework we are able to distribute large amounts of scenarios which have to be evaluated. This is described by Bergmann, Stelzer, and Strassburger (2012).

The results of parallel executed simulation experiments have to be collected and aggregated to performance indicators. This requires the model to have provided information necessary to estimate these values. We are again using CMSD-IM to hold result information. This enables the packaging of model, scenario description, and results in a single data exchange format.

The aggregated results are used by the controller to specifically generate new scenarios and can be presented to decision makers.

Based on the above discussed requirements, model predictive control or model-based decision support systems can be implemented (shown in Figure 3). The established cycle of generation, execution and evaluation of alternative scenarios (WIA) is identical to common simulation-based optimization loop. The advantage of this approach is the ability to consider current state information of the real system and the complex search-space description

4. DESCRIPTION OF ALTERNATIVES

While investigating applications in manufacturing system, we identified typical fields for decision alternatives as shown in Table 1.

Factors		
Job Schedule	**Shift Schedule**	**Machine Utilization**
Job-priorities	Human Resource Flexibility	Alternative Capacities
customer-priorities	Extra shifts	Express Deliveries
date-oriented priorities	Overtimes	Partial Deliveries
		External Capacities/ Out-Sourcing

Table 1: Common alternatives in manufacturing control

The set of available alternatives, of course, highly depends on the specific application scenario. Therefore some of the discussed alternatives may not be available or reasonable in every case.

After defining a set of common control options, we investigated ways for formally describing the possible alternatives. In relation to our previous work, we focused on how the Information Model provided by CMSD can be used for their representation.

4.1. Job Scheduling

The CMSD-IM provides a set of classes for the definition of schedules. A schedule in CMSD is reflected in the class Schedule, which consists of several ScheduleItems. In case of job scheduling, each ScheduleItem contains information when the assigned job has to be processed. Due the fact that more than one instance of the Schedule-Class could exist, CMSD is also capable of holding a set of different alternative schedules reflecting, for instance, different strategies of job-scheduling (e.g. job priorities vs. earliest due dates). A whole schedule contains a list of job references, tagged with time stamps for starting and finishing the associated job. The job reference can also link to an order reference, which is a list of jobs associated with a requesting party. With that it is also possible to prioritize customers.

With these capabilities of CMSD we are capable of defining alternative schedules based on job priorities, customer priorities, date-oriented priorities or combinations of all three.

Another common approach in job scheduling, also describable in CMSD, is to define the order of tasks by resource. In this case, a ScheduleItem in a schedule points to a defined ProcessStep instance and contains information about when it should be started. The ProcessStep-Class itself defines all required Resources like machines, workers or materials. For scheduling it is required to have previously defined all possible combinations of tasks and machines, which can perform these.

With this approach we can even further define the exact order of tasks (ProcessStep) of jobs, giving us full flexibility in representation of alternative job schedules. Beyond the task based scheduling, CMSD-IM allows the combination of ProcessSteps from different ProcessPlans. In this case, there are no limitations for scheduling of parallel machines. However, additional processing is necessary and later discussed in section 4.3.

4.2. Shift Scheduling

Workers in CMSD can have defined times, in which they are available for performing their operations. In a wide-range of applications the management of human resources is a major issue. Besides the task of scheduling times or shifts, it is also crucial to take into account the personal skills when assigning workers to process steps. The CMSD-IM provides several modeling approaches to address this. Every resource, including workers, is tagged with a CalendarReference. A Calendar defines a static list of times associated to Shifts or a reference to a ShiftSchedule. ShiftSchedules are more flexible and similar to the Schedule-Class used for job scheduling. There can be a set of alternative ShiftSchedules, but a resource can only be associated to one specific ShiftSchedule (via a CalendarReference).

A ShiftSchedule consists of a list of time-tagged ShiftReferences. The Shift-Class defines start and end times, breaks, and applicable days. In combination with a ShiftSchedule, it is possible to define the availability of resources on an arbitrarily level of detail. To perform a continuous scheduling of resources, it is necessary to define every time of activity (excluding breaks) as a Shift and merge them via a ShiftSchedule.

```xml
<ProcessPlan>
	<Identifier> ProcessPlan01_PartType01 </Identifier>
	<PartsProduced>
		<PartType> <PartTypeIdentifier> PartType01 </PartTypeIdentifier> </PartType>
	</PartsProduced>
	<Process>
		<Identifier> PP01_PartType01_Step01 </Identifier>
		<ResourcesRequired>
			<Resource> <ResourceIdentifier> Ma02 </ResourceIdentifier> </Resource>
			<AllowableSetup>
				<SetupDefinitionIdentifier> Setup_Ma02_PartType01 </SetupDefinitionIdentifier>
			</AllowableSetup>
		</ResourcesRequired>
		<OperationTime> <TimeUnit> minutes </TimeUnit> <Value> 35 </Value> </OperationTime>
	</Process>
	<Process/> ... <Process/>
<ProcessPlan>
```

Figure 4: ProcessPlan Definition (XML)

In some applications there are planned or potential deviations from the common Shift. In these cases we suggest to define a concurrent set of Shifts and group them into an alternative ShiftSchedule. In this approach it is only necessary to replace the schedule used for most cases with the special schedule.

With these capabilities of CMSD we are capable of representing arbitrary alternative shift schedules which can subsequently be used for evolution in the simulation.

4.3. Variation of Processes

Modeling the variation of processes is quite different to job or shift scheduling in CMSD. Processes are described by the ProcessPlan class which typically consists of a list of ProcessStep instances (see figure 4). A typical control option in the operation of a manufacturing system is the definition and selection of process alternatives for job or product types.

In case of machine failures or scheduled maintenance operations, alternative resources can be used to reduce the impact of such temporal bottlenecks. Like parallel machines, in CMSD-IM every possible process has to be pre-defined. If one or more tasks have to be changed in a process, a new ProcessPlan derived from the original one has to be defined. In this case, alternative process descriptions can be linked to (sub)-jobs or orders, changing the assigned EffortDescription. A similar line of action has to be performed if parallel machine setup is used. In principle, it is also possible to describe alternative resources in a ProcessStep instance, but they are not useful for scheduling of processes on parallel machines.

If the manufacturing system is operated by task-based scheduling, managing temporal variations of the material flow is also possible by using the Schedule-Class. As discussed in 4.1, the ScheduleItem-Class is able to describe the effort of tasks via ProcessSteps. Unfortunately CMSD provides no reference to an assigned job or order, which would be necessary to determine the effort for this.

To address this lack, we suggest using the CMSD provided property concept to attach a job or an order-reference to every ScheduleItem. Using this approach it is possible to assign every ScheduleItem to its proper job or order and determine the planned effort.

In summary, we can use CMSD to describe process alternatives in two different ways (direct modeling of alternatives in the ProcessPlan class vs. indirect modeling using the Schedule class). This gives the possibility to describe the usage of alternative capacities (compare table 1).

4.4. Additional Modeling

Partial or complete outsourcing of orders is modeled in the CMSD-IM using the Order or Job classes. Every instance is assigned to an executing party, which can be targeted to an external source. Further it is also possible to model costs of jobs, orders, resources, and tasks. By this, we can easily model express-deliveries of required materials or preconditioned orders and their financial impact on any cost functions.

5. GENERATION OF ALTERNATIVES AND COMPLEX SCENARIOS

While the previous section discussed the explicit description of alternatives to an existing model and how to reflect these alternatives in CMSD-IM, this section focuses on detecting further alternatives (possibly implicitly stored in the CMSD-IM) and composing more complex scenarios based on previously described alternatives and their variations.

The finding of alternatives beyond the explicitly modeled process definitions is based on investigating the predefined ProcessPlan instances. This is done by processing of the contained ProcessSteps and finding matches of output and input behavior, like consumed or produced parts or part types.

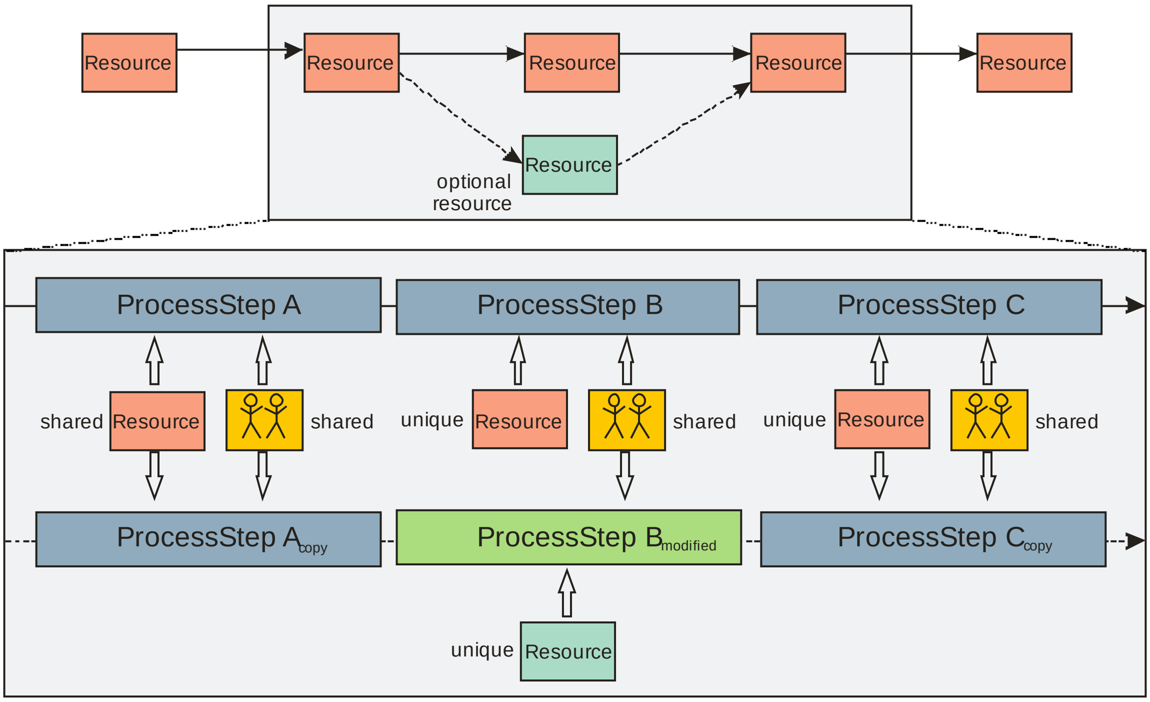

Figure 5: Modification of process definitions by exchange of required resources to create alternatives.

In our approach we assume that each ProcessStep reflects a task, which has to be performed by a defined set of resources. We also assume that tasks are exchangeable by another task exactly matching its input-output mapping. Based on these assumptions resources used to perform equivalent tasks are also exchangeable. Further information about required skills, setup-states and differing processing times can be determined by evaluating the associated resource references.

5.1. Iterating Process Alternatives

While generating schedules of jobs or human-resources are widely covered by commercial scheduling libraries, we here focus on generating alternative processes based on the modeled manufacturing systems. There are two prerequisite for our approach. At first the system and its current state is reflected in a CMSD-IM description. Second the manufacturing allows adaption, i.e., the system has a certain redundancy or unused resource potential.

Beyond the explicit modeling of varied processes, it is possible to determine alternative resources by processing the CMSD-IM model. Equivalent to the CMSD-IM approach, there are two stages for finding alternatives. The first stage is to process available ProcessPlan’s to find a matching of the produced parts or rather types of parts. In case of the ProcessPlan-level, they accord with a job. If there were explicitly modeled alternative processes for a job type, they would have matching input and output.

The next step is to use the determined information of equivalent resources to generate alternative process description (see top of figure 5). This requires a second pass, recursively processing ProcessPlans. For every existing ProcessStep there are two possible actions. If there is no alternative resource availed, the ProcessStep is kept untouched. Else, in the first pass an alternative ProcessStep was found. In this case, the processing is branched (forked) and an alternative ProcessPlan will be created, using the previously scanned ProcessSteps (see of figure 5). For the current ProcessStep we are able to create a new ProcessStep based on the alternative ProcessStep. To keep the consistency of the model, it is required to copy the required setup-states and worker skills. This transformation requires the absence of referenced human resources and machine-based skill.

If human resources have to be considered, the non-human resources are handled like described above, but the required human resources have to be compared by skills. If the skills of the worker performing the replaced operation are sufficient to perform the alternative task, we can reuse the references. If not, the human resources from the alternative ProcessStep have to be taken to the newly created alternative.

5.2. Dynamical building of Process Definitions

Another second practicable approach to determine alternative process definitions for exceptional situations or for analyzing possibilities is to use the current system load (for example: job situation) and to schedule these using all available operations. This requires the existence of an external scheduling library or software. The result of the scheduling process is transformed into a Schedule-Class instance of CMSD-IM. In this case it is necessary to use ScheduleItems based on ProcessSteps and tag them with the assigned job/order.

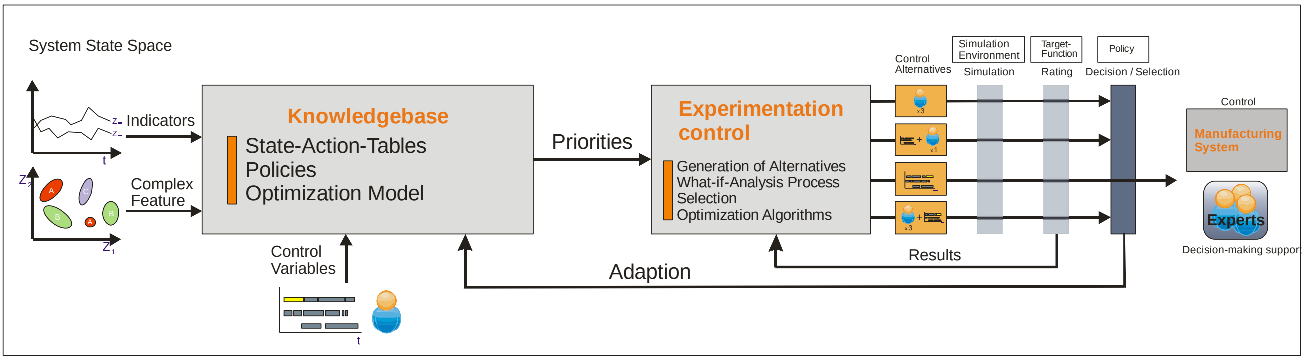

Figure 6: Description of algorithm used for selection of described alternatives and complex scenario (WIA) combination.

By processing a selected Schedule along the job reference it is possible to build an individual ProcessPlan and attach it to the job object by the JobEffortDescription.

5.3. Design and Generating of Complex Scenarios

For truly enabling MPC based on using discrete event simulation to evaluate different control options, a approach for generating complex scenarios is needed. Complex scenarios can contain combinations of the previously discussed alternatives.

In previous work we already discussed the automated model generation and initialization. Further we discussed several strategies to distributed execution of simulation experiments as a prerequisite for an efficient “what-if-analysis” process (WIA).

The missing link, like discussed in section 1, is the generation of complex alternative scenarios based on described decisions or control options. To enable a systematical experimentation, it was necessary to investigate the possibilities how to consistently model control or decision options (section 4).

In most scenarios it is required to apply a subset of alternatives to influence the system behavior. A controller and also decision support systems have to build complex scenarios, based on combination of available alternatives. In case of MPC, the controller has to do this automatically.

In our approach this is done by building complex scenarios by selecting subsets of the described decision and control alternatives. To define the subset, we are using a binary genetic encoding. To solve the problem that some alternatives, like scheduling or parameters are themselves iterable, we also use a genetic encoded description. By concatenation of the different encoding string, complex scenario descriptions are created. This string is used to generate a derivation of the base model of the manufacturing system, represented as a CMSD-IM description.

Using a binary genetic encoding also enables us to clip several dimensions of the search-space. To reduce the required iterations to find (semi)-optimal solutions we use a subset-selection algorithm, which starts with a minimal set of alternatives used for building complex scenarios. This subset is extended by unused alternatives, triggered by defined performance indices (figure 6). The used information model also enables us to determine the validity of the generated model.

5.4. Comparison and Discussion

To evaluate our approach to describe and automatically generate alternative control options or decisions and build complex scenarios, we implemented a CMSD-IM based symbiotic simulation environment.

The core of the test environment is an automated model generator, which is also able to initialize the generated models with the current system state. The model generator transforms a CMSD-IM description of a manufacturing system into an executable simulation model. This was recently discussed in previous work (Bergmann 2010; Bergmann, Stelzer and Strassburger 2011). The information flow is defined as shown in Figure 7.

We also implemented both ways of generating alternative process definitions to compare the impact on complexity and usability. Iterating the available process definitions and generating new ones based on compatible single-operations can generate a large amount of alternative process-definitions. This is amplified by the amount of alternative single-operations, especially induced by parallel machine-configurations or a high degree of redundancy. In this case we suggest focusing on explicit modeling and not using implicit information.

The second way of generating alternative processes (section 5.2) needs the presence of an external scheduling library which has to be provided with a set of process definitions. As previously discussed, alternative ProcessSteps can be regarded as parallel machine problems. This can lead to well-known problems regarding the solution complexity.

Our focus is to use a combinatorial iteration comparable to genetic encoding to build schedules. This only requires a mechanism to filter invalid schedules, which can be easily implemented.

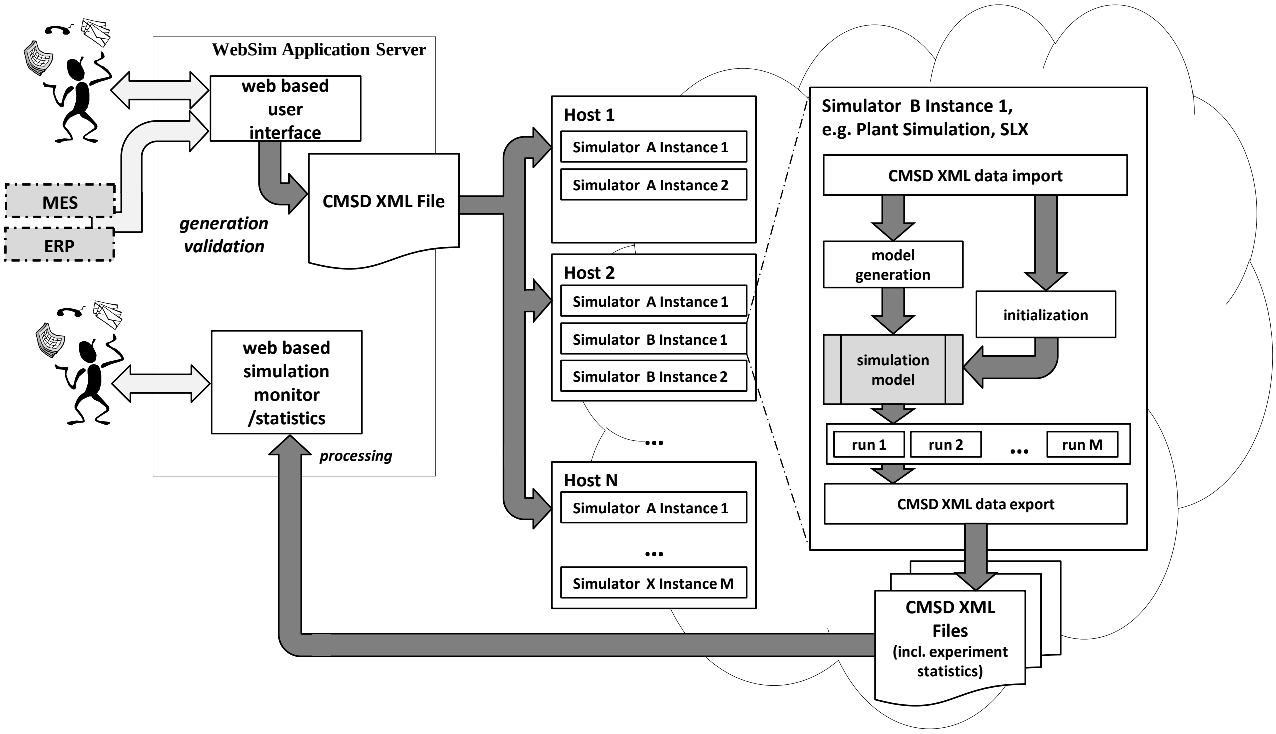

Figure 7: Information flow of the test environment for model generation and distributed execution.

6. CONCLUSION AND FUTURE WORKS

To enable model-based control, like MPC, for manufacturing systems, a way had to be found to resolve the problem of how to automatically iterate the search-space spanned by common control alternatives. In this paper we introduced an approach how to formally describe and model typical decision and control alternatives in the operation of manufacturing systems. We also presented a methodology for using implicitly stored information for detecting further control alternatives, which can be combined with explicitly modeled alternatives to complex scenarios. Using a Symbiotic Simulation System as a base, we enabled situation-based model-predictive control. The Symbiotic Simulation was implemented based on previously discussed work on automated model generation and initialization.

As shown in the discussion, our approach can also be useful to enhance model-driven decision support systems. The results of the work described in this paper enable us to investigate the MPC approach in real and larger scale manufacturing applications, which is a subject of future work.

Future work will also focus on intelligently handling the large amount of computation time needed to evaluate simulation experiments.

We also focus on applications for Exception based Manufacturing Execution Systems and the preventive evaluation of future scenarios in Early Warning Systems.

REFERENCES

Banks, J., Carson, J., Nelson, B. L., Nicol, D., 2000. *Discrete-Event System Simulation (3rd ed).* Upper Saddle River, New Jersey: Prentice-Hall, Inc.

Aydt, H., Turner, S. J., Cai, W., Low, M. Y. H., 2008. “Symbiotic simulation systems: An extended definition motivated by symbiosis in Biology”. *Proceedings of the 22nd Workshop on Principles of Advanced and Distributed Simulation*, 109–116. June 3-6, Rome (Italy).

Aydt, H., Turner, S. J., Cai, W., Low, M. Y. H., Lendermann, P., Gan, B. P., 2008. “Symbiotic Simulation Control in Semiconductor Manufacturing”. *Proceedings of the International Conference on Computational Science*, 5103:26–35. June 23-25, Krakow (Poland).

Bergmann, S., Strassburger, S., 2010. “Challenges for the Automatic Generation of Simulation Models for Production Systems”. *Proceedings of the 2010 Summer Simulation Multiconference*, 545-549. July 11-14, Ottawa (Canada).

Bergmann, S., Stelzer, S., Strassburger, S., 2011. „Initialization of Simualtion Models using CMSD“. *Proceedings of the 2011 Winter Simulation Conference*, 2228-2239. December 11-14, Phoenix (Arizona, USA).

Bergmann, S., Stelzer, S., Strassburger, S., 2012. „A new web-based method for distribution of simulation experiments based on the CMSD standard”. *Proceedings of the 2012 Winter Simulation Conference.* December 9-12, Berlin (Germany).

Davis, W. J., 1998. “On-Line Simulation: Need and Evolving Research Requirements”. In: Banks, J., ed. *Handbook of Simulation*. John Wiley & Sons Inc, 465-516.

Fowler, J. W., Rose, O., 2004. “Grand Challenges in Modeling and Simulation of Complex Manufacturing Systems”. *SIMULATION* 80(9): 469–476.

Heilala, J., Maantila, M., Montonen, J., Sillanpää, J., Järvinen, P., Jokinen, T., Kivikunnas, S., 2010. “Developing Simulation-based Decision Support Systes for Customer-driven Manufacturing Operation Planning”. *Proceedings of the 2010 Winter Simulation Conference*. December 5-8, Baltimore (Maryland, USA).

Hanisch, A., Tolujew, J., Schulze, T., 2005. “Initialization of Online Simulation Models”. *Proceedings of the 2005 Winter Simulation Conference*, 1795-1803. December 4-7, Orlando, (Florida, USA).

Henriksen, J. O., 1999. “SLX - The X is for eXtensibility.” *Proceedings of the 1999 Winter Simulation Conference*, 167-175. December 5-8, Phoenix (Arizona, USA).

Hotz, I., 2007. “Ein Simulationsbasiertes Frühwarn-system zur Unterstützung der operativen Produktionssteuerung und -planung in der Automobilindustrie” (A simulation based early warning system to support the operational production planning and scheduling in the automotive industry (in German)). Thesis (PhD). Otto-von-Guericke University Magdeburg (Germany)

Johansson, M., Leong, S., Lee, Y. T., Riddick, F., Shao, G., Johansson, B., Skoogh, A., and Klingstam, P., 2007. “A Test Implementation of the Core Manufacturing Simulation Data Specification”. *Proceedings of the 2007 Winter Simulation Conference*. 1673-1681. December 9-12, Washington, DC (USA).

Kletti, J., 2007. *Konzeption und Einführung von MES-Systemen*. Berlin:Springer.

Leong, S., Lee, Y. T., Riddick, F., 2006. “A Core Manufacturing Simulation Data Information Model for Manufacturing Applications”. *Proceedings of the 2006 Fall Simulation Interoperability Workshop*. September 12-15, Orlando (Florida, USA).

Siemens Product Lifecycle Management Software Inc., 2012. “Plant Simulation”. Available from: http://www.plm.automation.siemens.com/en_us/products/tecnomatix/plant_design/plant_simulation.shtml [March 2012].

SISO, 2010. “Standard for: Core Manufacturing Simulation Data – UML Model”. Core Manufacturing Simulation Data Product Development Group. Available from: http://www.sisostds.org/DigitalLibrary.aspx?Command=Core_Download&EntryId=31457 [February 2011].

VDI 3633-1, 2000. “Simulation of systems in materials handling, logistics and production - Fundamentals”. VDI-Society Production and Logistics. Berlin:Beuth Verlag.

Wang, W., Riviera, D. E., Kempf, K. G., 2007. „Model Predictive Control strategies for Supply Chain Management in Semiconductor Manufacturing”. *International Journal of Production Economics* 107:56-77.

AUTHORS BIOGRAPHY

SÖREN STELZER is a PhD student at the Ilmenau University of Technology. He is a member of the scientific staff at the Department for Industrial Information Systems. He received his diploma degree in Computer Science from the Ilmenau University of Technology. During his study he was working in the Neuro-informatics and Cognitive Robotics Lab of the Ilmenau University of Technology. After his study he worked in optimization of power plants in several projects. His research interests are simulation based optimization, model predictive control, artificial learning and discrete event simulation. His email is soeren.stelzer@tu-ilmenau.de.

SÖREN BERGMAN is a PhD student at the Ilmenau University of Technology. He is a member of the scientific staff at the Department for Industrial Information Systems. He received his diploma degree in Information Systems from Ilmenau University of Technology. Previously he worked as corporate consultant in various projects. His research interests include generation of simulation models and automated validation of simulation models within the digital factory context. His email address is soeren.bergmann@tu-ilmenau.de.

STEFFEN STRASSBURGER is a professor at the Ilmenau University of Technology in the School of Economic Sciences. Previously he was head of the “Virtual Development” department at the Fraunhofer Institute in Magdeburg, Germany and a researcher at the DaimlerChrysler Research Center in Ulm, Germany. He holds a Ph.D. and a Diploma degree in Computer Science from the University of Magdeburg, Germany. He is a member of the editorial board of the Journal of Simulation. His research interests include distributed simulation as well as general interoperability topics within the digital factory context. He is also the Vice Chair of SISO’s COTS Simulation Package Interoperability Product Development Group. His web page can be found via www.tu-ilmenau.de/wi1. His email is steffen.strassburger@tu-ilmenau.de.