

Discrete Events in a Continuous System

How to model influence factors on capacity in a complex assembly line environment

by

Johannes Hartwig

Supervisors:

Prof. Pål Davidsen University of Bergen

Prof. Mohamed Saleh Cairo University

Dr. Espacio Muñoz Pradas OSRAM Opto Semiconductors GmbH



Department of Geography
System Dynamics Group
Faculty of Social Sciences
Universitetet i Bergen

June 2011

Abstract

This study examines the influences of lot prioritisation, scheduling and line balancing on the productivity of a laser assembly line using a System Dynamics model. The handling of single production lots was implemented successfully into the simulation model and it could be shown that no hybrid approach is needed for combining System Dynamics and Discrete Event Simulation. Robust policies for prioritisation and scheduling were found; they depend upon local feedback policies as the composition of the product mix is not constant. For the same reason the application of line balancing methods was found to be challenging. The insights gained by this study were mostly qualitative and thus good practice for a System Dynamics project.

Acknowledgments

I would like to thank Ralph and Nòri for reading early drafts of my thesis. Thanks also to my two supervisors Pål Davidsen and Mohamed Saleh as well as Steinar Moen from Powersim AS. Many thanks also to Federica for supporting me during the conceptualisation phase. I would like to thank all my colleagues at Osram in Regensburg for providing a warm and stimulating working environment and namely my supervisor Espacio Muñoz and Martin Wachter who made this work possible after all. And finally my biggest thank goes to my family. Without them I would never have gone to Norway.

Contents

1	Introduction	1
2	The situation at Osram	2
2.1	The laser production at Osram	5
2.2	Planning horizons at Osram	8
3	Theoretical foundations and model framework	13
3.1	Literature review	17
3.2	The System Dynamics' modelling paradigm	20
3.3	The Discrete Event Simulation's modelling paradigm	28
3.4	Simulation models in the assembly line and semiconductor context .	34
4	The simulation model of the laser production	37
4.1	Description of the model	41
4.2	Modelling principle	43
4.3	Validation of the model	48
5	Results	60
5.1	Effect of prioritisation	61
5.2	Effect of scheduling	66
5.3	Effect of bottleneck removal and line balancing	73
5.4	Practical implications of the results	77
6	Conclusions	79
	References	82
A	Model documentation	94
B	Model equations	96

List of Figures

1	Reference mode of desired production	4
2	Process flow of products	6
3	Schematic representation of planning interactions	11
4	Decision foundation for capital investments	12
5	Basic dynamics of capacity acquisition	15
6	Stock and flow as basic elements of a System Dynamics model . . .	22
7	Basic system concept (Zeigler et al., 2000, p. 4)	25
8	Discrete events on a continuous time scale	31
9	Modelling principle	44
10	Basic building block of one process	46
11	Simulation results from parameter sensitivity test with scenario I .	53
12	Comparison of products in line in scenario I	54
13	Simulation results from parameter sensitivity test with scenario II .	55
14	Comparison of products in line in scenario II	55
15	Alternative runs week 5 scenario II	57
16	Completion time scenario II	57
17	Box plot completion time scenario II	58
18	Simulation results from parameter sensitivity test with scenario III .	59
19	Histogram comparison scenario III	60
20	Simulation results from change in prioritisation with scenario I . . .	64
21	Simulation results from change in prioritisation with scenario II . .	65
22	Simulation results of scenario I with and without scheduling	68
23	Simulation results of scenario II with and without scheduling	69
24	Average waiting time at process one	70
25	Products waiting at process one	71
26	Scheduling according to the Theory of Constraints	73
27	Different policies for bottleneck removal	76

List of Tables

1	Allocation of SD and DES to simulation model categorisation	33
2	Overview over the performed validation tests	49
3	Tested scenarios in the sensitivity analysis	52
4	Weekly production output in scenario I in lots	54
5	Weekly production output in scenario II in lots	56
6	Weekly production output in scenario III in lots	59
7	Selected scenarios for prioritisation analysis	62
8	WIP and finished lots in prioritisation testing I	63
9	WIP and finished lots in prioritisation testing II	64
10	Selected scenarios for scheduling analysis	67
11	WIP and finished lots in scheduling testing I	67
12	WIP and finished lots in scheduling testing II	68
13	Selected scenario for bottleneck analysis	75

1 Introduction

Capacity is one of the main constraints in manufacturing processes. The capacity of an assembly line is composed of the number of machines and operators; both are constrained by their physical processing abilities. Low utilisation of the resources lowers the profitability of the manufacturer as the fixed costs of capacity have to be distributed over a smaller quantity of sold products. Therefore companies strive for high capacity utilisation.

Capacity, on the other hand, is seldom as flexible as desired. The demand for products cannot be predicted with certainty and capacity cannot be augmented arbitrarily rapid. The acquisition of capacity can influence the competitiveness of a company significantly (Thillainathan, 1975). For a company with high capital costs for their plant equipment small fluctuations in their capacity can be worthwhile to examine. This is especially true for the semiconductor industry (Mönch et al., 2009).

Modern assembly lines are in most cases not restricted to a single type of product; mass-customisation allows for an economic reasonable individualisation of products (Boysen et al., 2006). Capacity fluctuations in this environment can originate from different scheduling of products, from prioritising the processing of one or a class of products over others or from inefficiencies in the assembly line set-up, often referred to as “line balancing”.

It is difficult to examine the factors influencing the overall capacity and hence the maximum production output in a complex manufacturing setting. Simulation is an appropriate method for doing so, as simulation models make assumptions explicit and are able to replicate results (Epstein, 2008). Furthermore, testing different scenarios on the real assembly line can be too costly.

OSRAM Opto Semiconductors GmbH is a laser producing company. They face similar problems like the whole semiconductor industry in their capacity planning process. Factors influencing the capacity like scheduling, prioritisation and line

balancing have not been examined in a satisfying manner yet. In this study the influences of those factors on line productivity are scrutinised with the help of a simulation model. The model is built with Powersim, a System Dynamics' software. System Dynamics was chosen because its proven ability of dealing with complex, non-linear systems (Sterman, 2000). The need for including discrete events was identified (Rose, 2000) and they were successfully implemented into the model.

Scheduling, prioritisation and line balancing (or bottleneck removal) are found to be largely dependant on the selected product mix. The composition of the product mix, however, is changing constantly. Therefore, fixed treatment rules are found to be inadequate. Instead, local feedback policies yield better results, are more flexible in their handling and easier to implement.

The thesis is structured as follows: chapter two describes the situation of the laser production department at OSRAM Opto Semiconductors GmbH in Regensburg. Chapter three lays the theoretical foundations of the simulation model; the generic model is described in chapter four. The most relevant findings are presented in chapter five and chapter six concludes with a summary and an outlook.

2 The situation at Osram

OSRAM Opto Semiconductors GmbH¹, a subsidiary of Osram GmbH, produces light-emitting diodes (LED), infra-red diodes and laser diodes and detectors for a broad range of industrial applications as well as consumer products. It has two production sites: one in Regensburg (Germany) and one in Penang (Malaysia). Osram is characterised by a very high level of professional specialisation, which directly results in a high level of organisational complexity (Dooley, 2002). The following description of processes refers to the frontend production of laser diodes, situated in Regensburg.

Modern laser diodes are optical semiconductors (Haug and Schmitt-Rink, 1984)

¹In the following Osram is used for OSRAM Opto Semiconductors GmbH.

and share the same basic production processes as “classical” semiconductors, described by (May and Spanos, 2006). However, there are some differences in process automatisation. The production volume is not as high as in electrical circuit semiconductor production and therefore manual process steps are more profitable than fully automatised processes. In addition to that the clean room requirements are less restrictive in the optical semiconductor industry and shop floor costs, which forms about 30% of total production costs in a “classical” semiconductor fab (Schömig, 2000) do not put that much economic pressure on space savings via automatisation. Furthermore, there are no industrial off-the-shelf solutions for many processing steps in the optical semiconductor industry compared to the integrated circuit producing semiconductor industry. The global demand for integrated circuits is much higher than for LEDs or lasers.

The production of lasers is highly customized: there are several specifications regarding the wavelength, size, emitting angle, material, power and operation mode (continuous wave vs. pulsed operation). Nevertheless, there are some standard products, ordered in higher quantities. As the amount of total production at the moment would commercially not justify an assembly line for each of those products, all are produced on the same line.

The laser market is highly research-driven; given product life spans are considerably short. The need for a constant product development results in a high percentage of test products. The prototypes are for economic reasons produced on the same assembly line on which shipped products are manufactured. So there is a constant trade-off between developing new products and selling existing products, given the restricted production capacity.

The knowledge of the production process can not easily be transferred to an eventual sub-contractor and even if so, this would not be desired. To attenuate the fluctuations resulting in non-constant ordering, policies of postponing orders are already in practice. This may lead to order cancellations, which is not only a problem of lost turnover but bears also the risk of losing customers in the long run. Therefore Osram seeks to increase their production capacity, as shown graphically

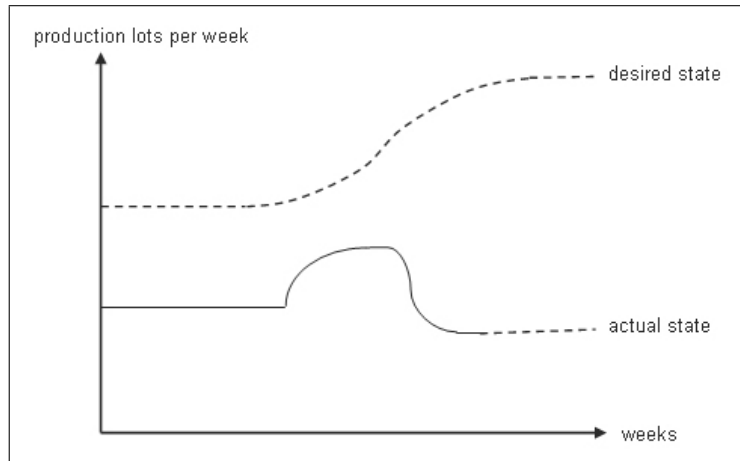


Fig. 1. Reference mode of desired production

in figure 1. The y axis shows the production output per week, measured in production lots. The x axis pictures the time. There has been an increase within the production for a shorter period of time, however, this was not enduring, though basic settings were not changed. Osram now wants to access influencing factors on production capacity in order to manage or even optimise them.

Three factors were identified that could have an influence on capacity and thus on overall production output. However, it was not clear how these factors could be assessed. Though no general agreement could be found whether or how much they exert an influence, some common beliefs about them existed.

Factors that are supposed to have an influence on overall productivity are:

- (a) **prioritisation:** Favouring one product at one process step over others will introduce longer waiting times for the other products. The same effect is assumed in overall productivity, leading to a loss in capacity.
- (b) **scheduling:** Setting up a processing order for the products is assumed to help in coordination and eventually in improving the overall production capacity.
- (c) **bottleneck removal:** Removing a clear bottleneck should increase production capacity and help in balancing the assembly line, which, in turn, should minimise the required resources for a given output.

Assessing these factors with a simulation model is the task breakdown for this thesis. The assumptions -or mental models- on those influences are to be examined and the question is to be answered whether there are general, robust rules deducible for these factors.

2.1 The laser production at Osram

The production of laser diodes and bars is defined by a high degree of internal complexity, as stated by (Größler, 2007, p. 203ff.). The degree of external complexity defines the degree of internal complexity in a positive relationship for manufacturing enterprises. This means that a very broad range of customers from divergent industries manifests the external complexity for Osram; and the different demands from the customers is the reason for a wide range of products, which, in turns, is responsible for a high degree of internal complexity.

An assembly line in the semiconductor industry is often divided into two major parts: the frontend production and the backend production (Mönch et al., 2009). The general processing of the wafers and probing takes place in the frontend, whereas the assembly and testing is located in the backend. Similarly, the chemical and physical processes and separation into laser diodes or bars and probing is also assigned to the frontend production at Osram. The backend assembles the diodes or bars and ships them to the original equipment manufacturers. One difference to the “classical” semiconductor industry exists: testing of single diodes or bars is also located in the frontend. Probing in the semiconductor industry refers to testing of whole wafers. This is not possible for lasers. They acquire their layers distinguishing them from LEDs after being divided into bars.

Frontend production itself is further divided into three major parts: epitaxy, front-of-line and end-of-line (EoL). Within the epitaxy the wavelength of the semiconductor emitter is determined and in the front-of-line different photolithography layers are applied. Within these two steps there is no major difference between LEDs and lasers and often the same resources are used for both types of semiconductors. The EoL is the part where the production process for lasers is unique.

The singulating of the whole wafer into diodes and bars and the mirror coating takes place in the EoL. In the EoL each product receives its characterisation and final optical inspection. This is also the part where manual processing is prominent. The EoL is the bottleneck of the overall laser production at Osram and this part of the manufacturing process is analysed in this thesis; all subsequent mentions of a bottleneck do only refer to the part of the EoL.

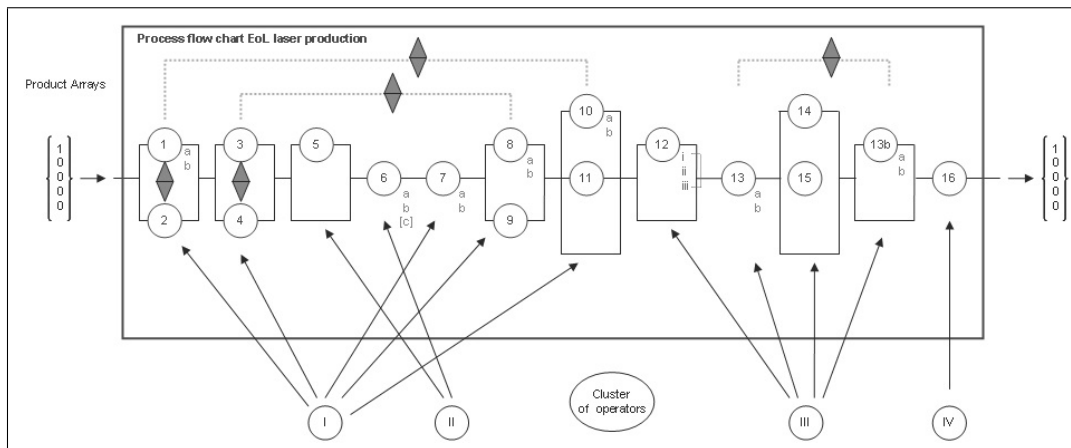


Fig. 2. *Process flow of products in the end-of-line.* The different products, represented by a product array, take different paths in their production process and use therefore different resources. The rhombi represent process conflicts, i.e. only one of the processes can be performed at the same time. The operators are grouped into four clusters. The small characters denotes duplicates.

Figure 2 shows a schematic overview of the production process with its many interactions. This figure gives an impression of the multiple possibilities of where the actual bottleneck may occur. The dotted lines with the rhombi represent process conflicts or re-entrant flows of products. Due to the long qualification times the operators are grouped into four clusters, denoted at the bottom of the picture. The small characters indicate that there are duplicates within a process (no single tools). Within each process various process steps are subsumed. The processing times of the different steps as well as of the different products for one step vary considerably. Furthermore, at some steps there are batch processes. All those factors have been identified as capacity loss factors by (Robinson et al., 2003). The complexity of the production process highly supports the observations made

by (Leporis and Králová, 2010), namely that detecting the bottleneck of production is not trivial at all.

The products are grouped into five families. The production paths are considerably different for each family. Figure 2 shows for every family different alternatives; only processes 6, 7 and 16 are passed by all products (but with different production times). Process 16 is the last process within the frontend production; afterwards the laser diodes or bars are shipped either to the backend production or directly to the customers. Manual handling processes are highly involved in process 1-3, 7-13 and 16.²

There is no unique measurement by which the productivity of the assembly line can be judged. The most common measure is lots; a unit that often corresponds to a 4-inch wafer. However, this number differs between product types as the size of the end product also differs. Other measurement units have been introduced in order to account for the yield and to facilitate the transfer to the marketing department, which is not so much interested in the number of wafers, more in the number of end products. For analysing the results I will therefore use the term production lot.

A main conceptual problem in the design phase was the determination of the level of detail (Fowler and Rose, 2004). In principle it is possible to include every detail of the assembly line into a simulation model. However, while it might be able with this approach to perfectly replicate all layers of detail, the ability for a rigorous analysis of policy options is reduced drastically due to the difficulty in defining the right leverage points; this happens if there are too many variables that can be altered. A clear analysis is strongly dependent on significantly reducing the complexity of the analysed system. Eventually, it is a question of being able to “combine information from a variety of sources into a single diagnostic or prognostic judgement” (Fischhoff, 1976).

On the other hand, a high aggregate level reduces not only the credibility of

²In appendix A on page 94 the exact process flows of the five products are listed.

the model as judged by the client group; it has also been shown that the inaccuracy arising from aggregation can lead to significantly wrong policy analysis and eventually the drawing of wrong conclusions (Jain et al., 1999). This is especially valid for the semiconductor industry (Rose, 2000).

2.2 Planning horizons at Osram

Linked to the question of accuracy is the definition of an appropriate time horizon for the simulation. I will show that this question is a question of defining the rank or tier of different planning horizons and their interrelationships.

There are three different planning procedures within Osram, all with a different and distinct purpose (compare also to (Reisinger, 2009) who described the planning horizons at Osram for the controlling department).

- (I) The first planning procedure tries to capture the production scheduling. The time horizon covers typically three months, based upon given customer orders. The purpose is to define the sequence of production and thereby to decide the delivery dates to the customers as well as to adjust the production to the current available capacity.
- (II) The second planning procedure is basically a definition of the budget for the next fiscal year. Therefore a detailed forecast over 12 months is made in order to define the pricing policy of each product according to its total projected demand and the financial resources of the company, which are to be aligned to the different business units.
- (III) The third planning procedure has a long term planning horizon: the demand for the next five years is forecasted. The fundamental need for this planning procedure results by virtue of capital commitment to production capacity and the long lead times for building up both the “hard” and “soft” parts of this capacity. Not only the delivery and customisation times for the machines are quite long, but also the time to recruit and teach workers

and engineers until they are productive typically exceeds the shorter-term planning horizon.

Those three planning procedures are interconnected. There is a substantial high complexity, both in terms of detail complexity, the microscopic system level, and dynamic complexity, the macroscopic system level (Bagdasaryan, 2010), as all three planning procedures normally require a deep and thorough understanding of their interconnections.

Each planning procedure, which I will name for convenience reasons the short-term (I), mid-term (II) and long-term (III) procedure has its own problems and difficulties, which unfortunately have strong impacts on the other procedures. So the interaction between the three planning levels is sufficiently high to exhibit fundamental estimation errors by ignoring the influence of one of them on the others. In addition to that the knowledge requirements at each stage are also sufficiently high to exhibit fundamental attribution errors by applying a too high level of aggregation.

A simulation model could assist in all three planning procedures. However, including all three procedures in one model at once would probably render the analysis impossible and, even if that was not the case, the involvement of the model's client would be much more crucial and difficult than in a simpler model (Jacobson et al., 2008). The more persons are involved in building a model, the more difficult it gets to achieve an agreement about its structure.

Focusing on procedure (I) would on the one hand address the problem that scheduling is currently -also due to the low level of automatisation- not possible with sufficient accuracy. A model picturing the assembly line in detail would on the other hand only represent a local optimisation of the production process, ignoring the long-term feedbacks of capacity requirements resulting from updated forecasts, captured in procedure (III). However, ignoring planning procedure (I) or subsuming it into a high aggregate model comprise the risk of an under- or overestimation of the capacity needed in the future for a certain production programme,

as a change in the forecast and thus in the production programme leads to very different capacity requirements. If, for example, the capacity could be increased by a different scheduling, then the increased capacity would allow for a higher mid-term planning (II). So there is clearly also an effect from procedure (I) on procedure (II).

Focusing on procedure (II) would allow for a better forecast and pricing, avoiding the Matthew effect, as a higher production forecast of one product would lower its unit price and making this product more attractive in terms of margin, leading either to an increase in marketing activities for this product or in lowering its unit price. Both consequences would fire the demand for this product, leading to a higher forecast, so that the initial (and maybe biased) advantage is accumulated over time. Ignoring this would not account for the longer-term effect, which would alter the capacity requirements, represented in procedure (I).

Focusing on procedure (III) seems to be the most promising area of analysis, as there is currently no strong focus on this planning horizon due to its inherent uncertainty and repeated negative experiences, whereas the decisions based upon this method have a high impact in terms of defining the need for acquiring capacity. However, as the diversity of customers is quite high, a planning procedure which would fully take the different industries into account would mean an enormous effort in modelling, and, what is more important, it is not clear what has to be done exactly in order to fulfil the production scenario, if no detailed model of the assembly line is included (e.g. acquiring machine α, β, \dots or employing x operators more).

The client in this study is the production department. Though the long-term planning (III) is the main driver for production planning, the production department is responsible for defining the necessary requirements. The department is responsible for not acquiring too many machines or operators and thus planning procedure (I) gains a lot of weight. Figure 3 shows how the different streams of planning from the marketing and development department (planning procedure (II)) merge together and set the foundation for the capacity planning in the production department (I).

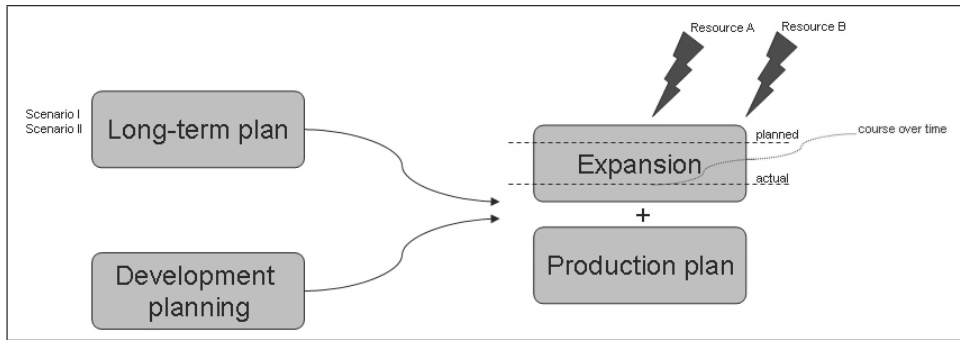


Fig. 3. Schematic representation of planning interactions. The planning from marketing (derived from the long-term plan (planning procedure (III)) and product development, both mid-term planning (II), merge together to the short-term capacity planning (I) of the production department

There is also another aspect of modelling the capacity requirements in more detail. In all planning a certain degree of uncertainty is involved; in fact it has to be allowed for it (Mula et al., 2006) for sustaining flexibility in the planning process. Due to the high degree of complexity the ignorance or degree of uncertainty is also relatively high as it is extremely difficult to meet all the particularities of different concepts, designed to address the specific problems arising at each planning method and to integrate them into an elaborate and practical planning procedure. The underlying question, whether it is more costly to allow for overcapacity than to allow for undercapacity in the long run in the light of uncertainty (Robinson et al., 2003) is therefore multidimensional and it seems nearly impossible to tackle it all-at-once with one problem structuring method (see (Badal, 2006) for an overview of problem structuring methods). Taking into account that strategic management decisions, like the decision on determining the future capacity, has to be very attentive towards weak signals of changes (Kreisler, 2005); reducing existing uncertainties in planning method (I) could as well assist in a more accurate planning of both other methods.

One illustration may serve here to illuminate how, for example, an interaction of planning procedure (I) and (III) could look like in practice, as shown in figure 4.

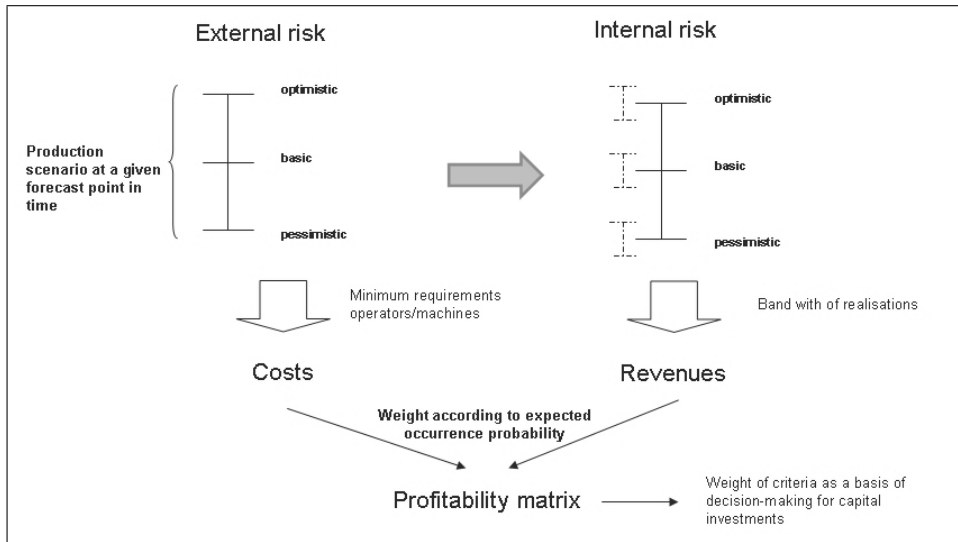


Fig. 4. Decision foundation for capital investments

The market defines the external risk (for which planning procedure (III) has to account for) and the production capacity represents the internal risk (for which planning procedure (I) has to account for).

The external risk is something the company has no influence on (Saleh and Myrtveit, 1999) and in the present case it would be mainly the long-term demand that is composed by the various industry demands for which an industrial or commercial application of the product exist.

The internal risk on the other hand is something Osram has an influence on. This risk results from uncertainties in the production process like machine down-times or operator absence; but there is also a deeper, more fundamental meaning of it. It is quite unlikely that a given product mix can be produced while using the full capacity of all processes. As the capital costs are quite high companies in the semiconductor industry nevertheless aim at doing so. However, as discussed above, the impact of some influence factors on the production capacity at Osram are currently unknown. So even if the product mix is set there is an uncertainty that this production programme can be fulfilled.

3 THEORETICAL FOUNDATIONS AND MODEL FRAMEWORK

Assuming that the factors mentioned above would have an influence on capacity: if the strength of those factors is unknown, then there is a risk that a disadvantageous use of one prioritisation set for example lowers the production output. So even if the nominal capacity is sufficient for producing a determined set, the risk of not fulfilling the desired output by a wrong choice in prioritisation (or scheduling) exists.

Only a combination of both may yield possible profitability measures, as shown in figure 4. Therefore it is not only important defining the external risk; the internal risk is as much of a concern to effectively steering the future development of a manufacturing company. A model adequately representing the internal risk (and assisting within the short-term planning process (I)) can thus also assist in longer-term planning processes.

3 Theoretical foundations and model framework

Production is usually constrained by capacity, costs and quality. One can only produce as much as capacity allows for. Capacity is a constraint because of costs and quality. If costs are not constrained, one could build up excessive capacity to fulfil every customer demand.

But higher costs result in higher prices and higher prices are rendering the products less attractive for customers. Bad quality, on the other hand, lowers also product attractiveness, but it has also an effect on capacity. If quality drops more rework has to be done, or, even worse, more scrap is produced, increasing the costs for the good products. So the capability for producing valuable goods is lowered.

Capacity is thus a central component in a company's strategy. (Forrester, 1968) has pointed out the strong effects capacity expansion can have on the development of a company, more than external influences like market share or fluctuating customer demands. (Thillainathan, 1975) has well summarised the importance of capacity acquisition:

3 THEORETICAL FOUNDATIONS AND MODEL FRAMEWORK

Capacity acquisition can have significant effects on the growth and stability of a firm and it is possible that some of the fluctuations or even collapse of the activities of a firm are nothing but the manifestations of ill-designed capacity acquisition policies. (...) It is no exaggeration to say that capacity planning has an overriding effect on all the other functions related to production and (...) capacity planning would be on top feeding into such sub-systems as Sales (...) and Distribution. (Thillainathan, 1975)

A capacity constraint may be expressed by volume limitations of a machine or by the amount of operators working at the assembly line. Some of those constraints are fixed restrictions in a short term view. A new machine can not be acquired, delivered and customised immediately, and, in many cases, a new operator must run through qualification stages before he is able to work productively. However, in a long term view those constraints are not fixed any more. Unfulfilled orders are piling up in the backlog and an increase in backlog as well as expectations about future sales lead to pressure to extent production capacity. This can be regarded as a feedback loop: actual production capacity and desired production capacity form a gap which leads to capacity acquisition which increases production capacity. So in the long term there is an adjustment effect active that balances the actual and desired production capacity.

While this feedback effect can be ignored in a short term production planning the limit for the production capacity may be variable within a certain range. The resources needed for a product in terms of machine and operator time may vary. Therefore it is possible to realise different production sets within a given environment. That said one is able to trade off different production scenarios against each other in order to gain maximal profit out of the given resources.

However, one feedback effect is effective also in the short run: the overall production capacity influences the forecast as well. If there are excess orders that cannot be fulfilled due to the restricted capacity, customer orders will be piling up in case they are not cancelled. So, future capacity requirements may seem to

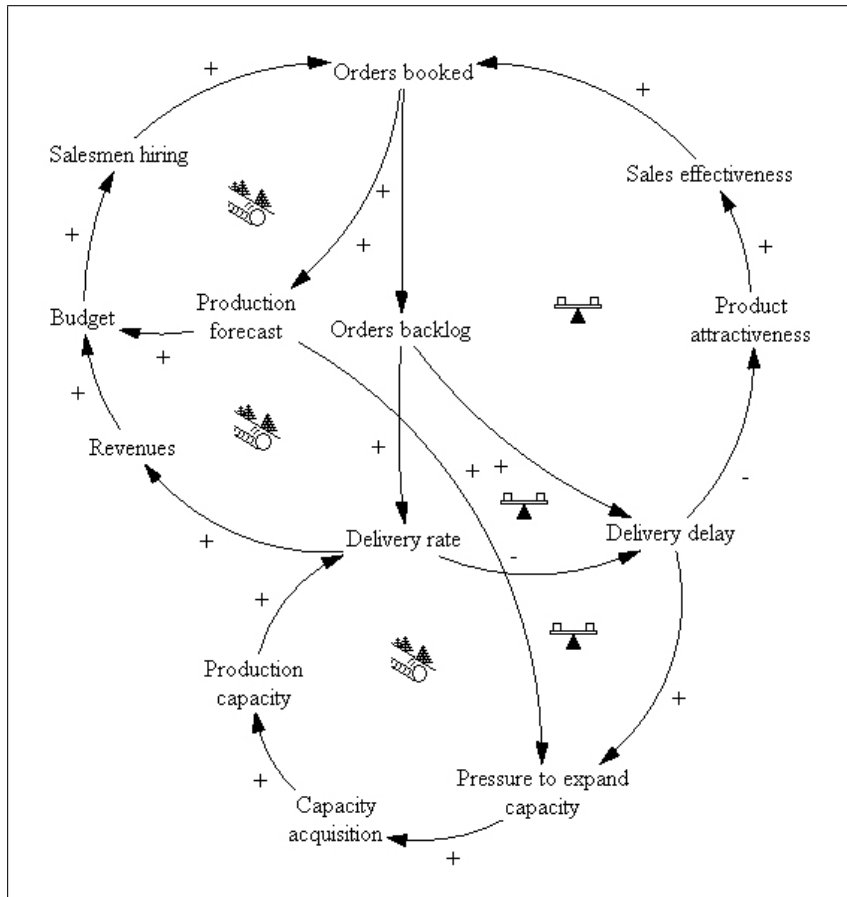


Fig. 5. Causal loop diagram, showing the basic dynamics of capacity acquisition, partly adopted from (Forrester, 1968). The balance indicates a balancing loop and the declivity a reinforcing loop

be much higher because of the delayed order fulfilment. Those effects, as shown in figure 5 have been discussed in the last chapter at the example of planning horizons at Osram.

In order to come to an appropriate measure to judge on improvements of an assembly line one has first to structure the problem. There are some problem structuring methods that has been successfully applied within an assembly line environment. Almost all of those methods are “hard” methods, meaning that they seek for a quantification of results. Although it might not be surprising that “soft”

3 THEORETICAL FOUNDATIONS AND MODEL FRAMEWORK

methods are hardly applied in the context of assembly line analysis, there are still huge differences within the “hard” approaches. These differences are not mere philosophical, but cover questions like the role of data accuracy, the influence of stochastic elements and the desirableness of finding an optimal solution to a given problem.

Two main methods can be distinguished within the “hard” problem structuring methods: analytical methods and numerical methods. While both methods try to picture essential features in a mathematical model they differ in the way the model is developed and solved.

Analytical methods seek for a single, optimal solution to a given problem with algebraic means. For the price of analytical clarity and mathematical elegance it is often unavoidable to highly simplify the problem in order to achieve such a solution.

Numerical methods often involve either some kind of computer simulation model of the problem or an algorithm procedure. This enables to include much more realistic assumptions. However, this also includes the risk of overloading the model with too many variables which may hamper a thorough analysis of the different factors. It increases the possibility of contradicting assertions or may even sometimes inhibit finding an optimal solution in cases where this is desired.

The inclusion of more realistic assumptions was the starting point for this study: albeit one may be able to derive a certain production set that seems to satisfy given resource constraints, there may be other influencing factors on production capacity within a complex production environment; factors that determine the production order like scheduling and prioritisation or that have a direct influence on the degree of capacity utilisation like line balancing. Different beliefs about those influencing factors exist at Osram; however, it has not been possible to challenge or test those assumptions so far. This study aims at gaining a deeper understanding of those factors and the influence they may or may not have on overall production output.

3.1 Literature review

This study is a combination of System Dynamics (SD) and Discrete Event Simulation (DES) in the field of assembly line modelling. The foundations of SD can be found in (Sterman, 2000), those of DES in (Banks et al., 2005); cross-studies comparing both has been conducted e.g. by (Sweetser, 1999; Morecroft and Robinson, 2006; Tako and Robinson, 2008; Özgün and Barlas, 2009). SD could be assigned to the DESS formalism, as described by (Zeigler, 1976).

Differences between SD and DES lead to different focuses in the analysis of systems (Bertrand and Fransoo, 2002). SD is more used in strategic modelling (Chahal and Eldabi, 2008), whereas DES modelling is more active at an operative basis (Semini et al., 2006). However, modelling production systems with SD is a missed opportunity so far (Baines and Harrison, 1999).

Research in the semiconductor industry with SD has been carried out for example by (Gonçalves, 2003; Bezemer, 2003; Chen and Jan, 2005; Wu, 2007), but only in the field of supply chain modelling or the industry as a whole. The methodological value of SD in Operations Management has been shown by (Größler, 2007; Größler et al., 2008), but SD is still under-represented in assembly line modelling and decisions on an operational level (Listl and Notzon, 2000; Godding et al., 2003; Filho and Uzsoy, 2010); this is more a domain of DES (Baines and Harrison, 1999). The reason why research in the SD is more focused on a larger system's division like supply chains can be found in the belief that a system's behaviour is more determined by dynamic than detail complexity (Senge, 1990). Too many details in a larger system are regarded as being disadvantageous (Jain et al., 1999). However, there are situations where simple models fail (Rose, 2000), especially in semiconductor manufacturing, which is characterised by a high production mix and low production volume (Johnzén, 2009).

The need for taking discrete events into consideration can be derived partly from studies that compare SD and DES, but also from the complexity of the semiconductor manufacturing chain (Jain et al., 1999; Schömig, 2000; Mason and

3 THEORETICAL FOUNDATIONS AND MODEL FRAMEWORK

Fowler, 2001). For including discrete events into a SD, another formalism, the DTSS (Zeigler, 1976), is used here. The timestep plays a crucial role in it. It can be seen as an extension of the classical description of a SD model (Kampmann and Oliva, 2008), which was also made by (Vaneman and Triantis, 2003) in a similar way.

This procedure avoids the need for building a hybrid model. Hybrid modelling (Rabelo et al., 2005; Venkateswaran and Son, 2005; Helal et al., 2007; Hao and Shen, 2008) combines SD and DES via communication interfaces; so two separate models are required. With the method presented here in this study this is not necessary. The modelling is not multi-formalistic like the combination of DEVS and DESS via coupling, which was theoretically shown by (Zeigler et al., 2000; Vangheluwe et al., 2002). The imprecision of a DTSS compared to a classical DEVS is abated by choosing a sufficient small timestep. (Dvergsdal, 2006; Linge, 2007) have used a similar approach in healthcare modelling, but this is the first study in the area of manufacturing simulation.

Three research questions form the hypotheses for the practical simulation study at Osram: the effects of *a*) prioritisation, *b*) scheduling and *c*) bottleneck detection and removal on overall production output. All three are related to capacity loss factors in semiconductor manufacturing (Robinson et al., 2003), and the former two are thought to be means to increase capacity without monetary investments.

Prioritisation has been discussed by (Chik et al., 2004; Crist and Uzsoy, 2010), but it is the least investigated issue of the three in operations planning. However, it gained some attention for setting up dispatching rules in the semiconductor industry (Mason and Fowler, 2001).

Scheduling is a topic embedded in a much broader context, e.g. (Wu, 2005; Kogan, 2006; Mönch et al., 2009); it is viewed as a major issue requiring thorough investigation and coordination (Kádár et al., 2004). In the literature a dominance of linear programming methods is dominant (Potts and Kovalyov, 2000); simulation is not so prominent as it does not as easily support the endeavour for finding

3 THEORETICAL FOUNDATIONS AND MODEL FRAMEWORK

an optimal solution.

Line balancing has attracted the most attention in the literature so far (see, for a classification of line balancing problems and a comprehensive literature review (Boysen et al., 2006); (Mahayuddin and Tjahjono, 2010) remark in their classification of manufacturing simulation papers a dominance of line balancing issues). (Falkenauer, 2005) states that while most line balancing models are mathematically sound and sophisticated, they often miss a real-life implementation.³

The literature on the aforementioned topics is coined by formulating mathematical, quantitative models. What is missing are qualitative insights. An exact measure of the factors influencing assembly line outputs is difficult and (Bartholdi et al., 2009) have shown that even simple discrete manufacturing models can exhibit unpredictable behaviour. The task of this thesis is twofold, on a theoretical and on a practical basis.⁴

Theoretical contribution of this thesis This thesis shall give an example of inserting discrete events in a SD model in a manufacturing environment. The need for it has been identified in the literature; but the realisation has been made with combining two methodologies so far. With the presented framework here it is possible to avoid multi-formalism modelling.

Practical contribution of this thesis For Osram qualitative insights regarding the effects of prioritisation, scheduling and line balancing are important in order to judge on future decisions regarding extensive use of quantitative, mathematical models. These qualitative insights shall be gained by this study.

The theoretical foundations for the model are presented in this chapter; the next chapter discusses the practical implementation of the theoretical findings

³Adding to this most of the presented papers in line balancing focus solely on machine constraints, the resource constraints that are established by the operators is often neglected (Baines et al., 2004).

⁴In this study uncertainty is not considered for reasons of simplifications. (Mula et al., 2006) lists models for production planning which takes uncertainty into consideration.

and in the subsequent chapter the findings regarding the qualitative insights are presented.

3.2 The System Dynamics' modelling paradigm

For defining a client's problem it is essential to come to the core of the "real" problem. It is thus mandatory to get first acquainted to the system and derive the essentials of the problem. This is especially important in order to not focus on minor (or superficial) problems that are embedded into major (or deeper) structures, because a sub-optimisation of an embedded problem may cause an overall system to perform worse afterwards. Therefore it is helpful to look at the system as a whole.

This approach can be called a systemic approach, which is sometimes outlined as systems science. There are a variety of systems science methods which mostly have evolved after the Second World War, like *a)* Cybernetics, *b)* General Systems Theory, *c)* Systems Intelligence, *d)* Systems Thinking and *e)* System Dynamics. Though all of them have in common that they try to tackle a system from a holistic point of view, only System Dynamics can be called a "hard" approach in the sense that it is able to quantify the behaviour of the system.

System Dynamics (SD) as a systems science discipline⁵ could also be regarded as a problem structuring method. It assumes that it is not so much a question of missing information for solving a problem, rather the mental capacity of the people within the system that is limited and, especially, different and incomplete point of views of what are the main drivers for the undesired behaviour.

This is often referred to as the mental models: "A mental model of a dynamic system is a relatively enduring and accessible, but limited, internal conceptual representation of an external system (...) whose structure is analogous to the perceived structure of that system" (Doyle and Ford, 1999). The mental models

⁵It could be questioned whether or not it is justified to call it a discipline while there are no formal boundaries between the different streams of systems science; indeed they exist in parallel without taking much notice from each other.

3 THEORETICAL FOUNDATIONS AND MODEL FRAMEWORK

of different persons have usually divergent assumptions: $\dot{x}(t)$ may be for person i a result of a_i, b_i, \dots , resulting in his mental model $\dot{x}(t) := f(\hat{a}_i, \hat{b}_i, \dots)$ with \hat{a} representing the estimated or perceived value of a . For person j the result or mental model is $\dot{x}(t) := f(\hat{a}_j, \hat{b}_j, \dots)$. $\dot{x}(t)$ is the observed behaviour of a system $s = \dot{x}(t)$ from person i, j, \dots and $k = a, b, \dots, n$ are the influencing factors in the system s . It may well be that $\sum_a^n k_i \neq \sum_a^n k_j$, i.e., that there are different assumptions regarding factors that play a role in a system's behaviour. Furthermore, it could be that $\hat{a}_i \neq \hat{a}_j$. And finally, of course, there could be disagreement on the strength of each element of k .

SD aims to reveal those mental models by making them explicit, in order to achieve a common base of all stakeholders of the system relevant to the problematic behaviour. The ultimate goal is to change those mental models so that a desired system behaviour via a change of structure could be achieved. This is then done by finding a common ground for defining $f(\hat{a}_{ij}, \hat{b}_{ij})$.⁶

Making a mental model explicit could be done in different ways. One way is to draw a Causal-Loop-Diagram (CLD), as shown in figure 5 on page 15. The balance denotes a balancing loop and the declivity denotes a reinforcing loop (for a definition refer to (Sterman, 2000)). However, there are some problems related to CLDs. (Schaffernicht, 2010) summarised the multiple problems that arise with the use of CLD, such as *a*) lack of precision, *b*) loss of distinction between stocks and flows, *c*) wrong labels of polarities and *d*) a pure graphical representation, whose interpretation in contrast to a simulation depends on the interpreter.⁷ To summarise this critique: a CLD may be good for communication purposes, but it alone falls short of the main purpose of a SD study: the construction of a simulation model as the testing ground for hypotheses or mental models. A SD model is the structural explanation or hypothesis of an observed phenomenon or behaviour.

⁶It should be noted that within the Operations Research (OR) community this change of mental models, though not explicitly labelled as such, is recognised. (Eden and Ackermann, 2006) stress the fact that “[i]n the end, all OR [Operations Research] is about changing minds and actions of people, not organisations”.

⁷In addition to that (Güneralp, 2004) found in his analysis that in a second order system (and thus quite a simple system structure) the relative location of feedback loops can be a determinant of the kind of behaviour that arises out of the structure.

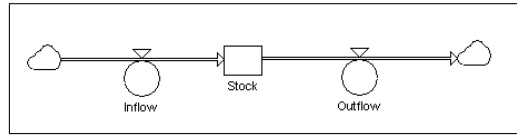


Fig. 6. Stock and flow as basic elements of a System Dynamics model

Another way of making a mental model explicit is a Stock-and-Flow-Diagram (SFD), as shown in figure 6.

The definition of a stock is

$$Stock(t) = \int_{t_0}^T [Inflow - Outflow] dt + Stock(t_0).$$

The flows affecting a stock could be summarised as one net flow (Wagner, 2004), giving the definition

$$Net\ flow = \partial(Stock)/dt.$$

(Sterman, 2000) gives a detailed explanation of this graphical representation. All big SD simulation software (Powersim, Vensim and iThink/Stella) use a SFD to display the mathematical foundation of a SD model: differential equations.

(Kampmann and Oliva, 2008) describe a SD model as a set of differential equations of the form

$$\frac{dx(t)}{dt} \equiv \dot{x}(t) = f(x(t), u(t)),$$

with $x(t)$ as a column vector of n state variables $x_1(t), \dots, x_n(t)$ (those are the levels or stocks, see also figure 6), $u(t)$ as the column vector of p exogenous variables $u_1(t), \dots, u_p(t)$, f as the vector function and t as the simulated time.

In the light of this definition it is possible to allocate SD to one of the modelling formalisms, as proposed by (Zeigler, 1976). The correspondent modelling formalism would be the Differential Equation System Specification (DESS).

3 THEORETICAL FOUNDATIONS AND MODEL FRAMEWORK

A DESS is described as follows:

$$DESS = (\chi, Q, q_0, \Upsilon, f, \lambda)$$

DIFFERENTIAL
EQUATION
SYSTEM
SPECIFICATION

where

χ represents the set of input values;

Q is the set of internal states;

q_0 is the initial state;

Υ is the set of output values;

$f : Q \times \chi \rightarrow Q$ is the rate of change function and

$\lambda : Q \rightarrow \Upsilon$ is the output function.

The constraints of a DESS are given by *a)* the Lipschitz condition for f and *b)* the condition $\chi, Q, \Upsilon \in \mathfrak{R}$.⁸

Defining a SD model as a model of differential equations allows to consider a long-term view as described in the precedent chapter. Furthermore, the portrayal of this differential equation system via a SFD diagram as a less technical representation assists in communicating the structure to the stakeholders of the system: “System Dynamics is a tool to communicate and compare the result of different points of view of how reality is perceived” (Frandsberg, 2003).

However, there are some difficulties with the assignment of a DESS formalism to a SD model:

(a) *what are the input values χ ?* For a SD model one may claim that these are the

⁸The rate of change function f has to be a continuous function. With this condition it is not possible to include discrete events.

3 THEORETICAL FOUNDATIONS AND MODEL FRAMEWORK

parameters that change the stocks, such as auxiliaries, flows and constants⁹; however, one may also claim that these are the constants or external auxiliaries (or, more specifically: all those parameters that affect the simulation outcome by no endogenous loop). This is closely related to the next question:

(b) *what is the initial state q_0 ?* While, at a first glance, this might seem to be quite obvious: the values of all parameters at the beginning of the simulation, this definition has some implications on how you look at a SD model: one claim of the SD community is that structure defines behaviour. But what is the structure? Is it the stock-and-flow diagram? Is it the stock-and-flow diagram with its initial values?¹⁰ Note that different initial values could result into a totally different behaviour of the system, as shown in the analysis of (Saleh and Davidsen, 2001). Here the SD community misses a straight definition which would facilitate both the analysis of the simulation models and the communication to other members of the Operations Research community. Connected to these two questions is the next one as well:

(c) *what is the output of a SD model?* Also this is not well defined. It could be a) defining the output for every Δt like $\Upsilon = \chi_2$, while χ_1 are the values at the beginning of the timestep and χ_2 are the values at the end of the timestep. It could also be b) the output graphs of the main stocks. Defining the output in a narrow way would mean that for every timestep there is the possibility of a Δ describing the deviance from the simulation results to actual system's behaviour; defining the output in the latter way gives another definition problem: one has to define the main stock values¹¹ which are more relevant in judging on the validity of a model. This would establish a hierarchy, but it could simplify validation tests as not all stocks have to be considered in these tests. However, option b) has one big blemish: the introduction of the timestep. This is discussed in more detail below.

⁹Remark that in a SD model the stocks represent the state of the system and they can only be changed by flows, which can be influenced by the stocks, but also by auxiliaries and constants. The flows would then be the highest order of input values, as these are the values that are calculated after the auxiliaries, just before the new state of the stocks are calculated. The flows are the only way to influence the stocks.

¹⁰This would mean that other initial values would represent another structure.

¹¹And auxiliary values for that respect.

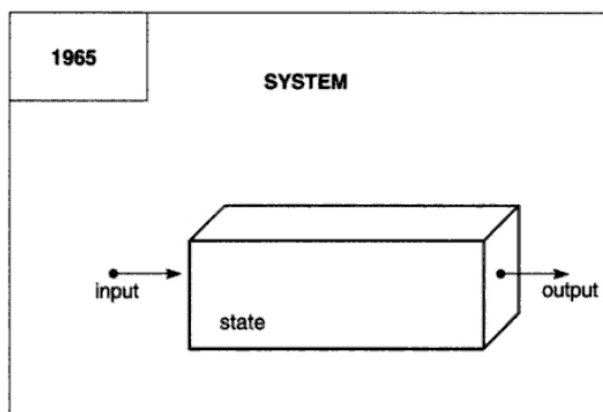


Fig. 7. Basic system concept (Zeigler et al., 2000, p. 4)

The idea of a structural explanation for behavioural symptoms (Schwaninger and Hamann, 2005) is not a unique claim of the SD community, as the following quote from the Operations Research community shows:

System theory distinguishes between system *structure* (the inner constitution of a system) and *behavior* (its outer manifestation). (...) Knowing the system structure allows us to deduce (analyze, simulate) its behavior. Usually, the other direction (inferring structure from behavior) is not univalent - indeed, discovering a valid representation of an observed behavior is one of the key concerns of the M&S enterprise. (Zeigler et al., 2000, p. 3f.)

The figure 7, taken from (Zeigler et al., 2000) may also be valid for a SD model. This leads to the question exposed by (Vázquez and Liz, 2007): “In what sense can the structures postulated by SD models be assumed to exist objectively in reality?”.¹² This remains an open question for the SD community.

¹² (Fritz, 1983) discusses the problem that language imposes in addition on the construction of SD models. This is not hindered by the mathematical foundation of SD models, as mathematics is also a way of communication. He further states that “judgement and interpretation are closely linked with perception, linguistic conditioning and ideology”; in the light of this statement defining the structural foundation of a system may become arbitrary. And, as (Dent, 2001) states a SD model is not able to reflect a multiplicity of perspectives on the same phenomenon; some representation of reality that might be unquestionable for someone might be deeply challenged by someone else.

3 THEORETICAL FOUNDATIONS AND MODEL FRAMEWORK

Despite of those open questions the use of SD as a structural theory (Schwaninger and Hamann, 2005) is not challenged here. The structure of a model is usually approved by its stakeholders and those questions arise partly also within a non-graphical representation of a system with differential equations, so it is not dependant on the formal depiction of the system. A model remains a simplification of reality (Sterman, 2002), no matter how sophisticated it might be.

An analytical solution is only possible for the simplest simulation models. As these models have only very limited explanation strengths all SD programs calculate a numerical solution for the differential equations as a default setting. This is done by the introduction of a timestep. This timestep is a partition of a simulated time interval and the intermediate points of the calculation algorithms. Standard SD software use the most common numerical solution algorithms for differential equations, which are Euler or Runge-Kutta of higher order (for a detailed explanation refer to (Sterman, 2000)). The partition density of those timesteps can be chosen by the user. However, the representation of a SD model is then not a pure DESS any more.

This conflict has already been addressed. (Vaneman and Triantis, 2003) give another description of a SD model: “System dynamics models can be characterized as continuous at discrete points in time, thus the model values changes smoothly, but are only accessed at specific time steps”. They describe a SD model as a dynamic, causal and closed system:

$$y_{jt} = f\{t - t_0; x_{it_0}; x_{itd}; y_{j(t_d-t_0)}\};$$

y_{jt} is the j th output from action in the interval $[t_0, t_d]$, where t_d represents some intermediate time with $t_0 < t_d < t$, allowing for some input x_i at time t and yielding in output y_j at time t . Remark here the difference to the description of (Kampmann and Oliva, 2008), who do not introduce an intermediate time t_d . This time t_d gives discontinuity to the system. However, it resolves the problem of a production line, that is dependent on extraneous inputs at various times t_k ,

3 THEORETICAL FOUNDATIONS AND MODEL FRAMEWORK

because a production line can be characterised by a set of input vectors x over a given time horizon $[t_0, t_d]$ that gives some output y ; in absence of inputs no output is produced¹³. This means that even for $x_{t_0} > 0$, after a long enough time, the production line is in constant equilibrium, e.g., it produces no further output¹⁴. So omitting further inputs at time t_d gives only a very limited picture of the dynamics of a normal assembly line. A “pure” interpretation of a SD model in terms of differential equations becomes -at least in context of an assembly line- questionable.

In the light of this discussion most SD models could be classified into another formalism, the Discrete Time System Specification (DTSS).

A DTSS, as described by (Zeigler, 1976), is defined by:

$$DTSS = (\chi, Q, q_0, \Upsilon, \delta, \lambda)$$

DISCRETE
TIME
SYSTEM
SPECIFICATION

where

χ is the set of input values;

Q is the set of internal states; in a SD model these could be the stocks and auxiliaries at the beginning of a timestep¹⁵; see also the discussion above;

q_0 represents the initial state; this merely means Q at t_0 ;

Υ is the set of output values;

$\delta : Q \times \chi \rightarrow Q$ is representing the single step transition function; for a SD model this could be the change in auxiliaries and stocks as described above and

$\lambda : Q \rightarrow \Upsilon$ is the output function.

¹³This fulfils one production axiom: $y_t \notin P(x_{t-t_0}; y_{t_d-t_0}) = 0, y_t > 0$, so there is no “free lunch” because it is not possible to produce outputs at time t when there is no input in the time interval $[t, t_0]$ (Färe and Primont, 1995).

¹⁴A production line is only in a constant equilibrium, when $0_t \in P(x), \forall (x_i \vee y_j) \mathfrak{R}_+^N$ (Vaneman and Triantis, 2003).

¹⁵While the focus of SD lies in the observation of the stocks which are claimed to be the physical representation of the system, auxiliaries that change influenced by the stocks and that have a connection to flows have, in a pure interpretation of the DTSS formalism, also to be \in of Q .

3 THEORETICAL FOUNDATIONS AND MODEL FRAMEWORK

The main difference in the DTSS formalism to the DESS formalism is the definition of f respectively δ . Defining a SD model with a DTSS formalism gives a further advantage: some standard functions in SD programs like the if- or the pulse-function do not contradict this formalism.

One aspect in this discussion is worth stressing: the concept of time. (Ossimitz and Mrotzek, 2008) remark that there is some confusion about it in the SD community: "Despite of its central role, both theory and practice of SD does not much bother about how to model time". It is not clear whether time is a continuous or a discontinuous phenomenon. Though (Forrester, 1961) wrote: "Real systems are more nearly continuous than is commonly supposed" some sensitivity tests are discontinuous by their nature and taking a discontinuous testing function for validating a purely continuous system seems not to be too consequent.

Different point of views about the concept of time led to great extent to the development of another simulation technique over the decades, mostly in parallel to SD: Discrete Event Simulation (DES). (Wolstenholme, 1983) stated: "any natural phenomena is, of course, a mixture of discrete and continuous relationships" and it cannot be stated in general which perception of time is the most appropriate one. The idea of a discontinuity of time goes hand-in-hand together with the focus on single items in a DES system (see discussion below).

3.3 The Discrete Event Simulation's modelling paradigm

SD and Discrete Event Simulation (DES) models often do not share the same world-view (Morecroft and Robinson, 2006). The two most striking elements are interconnected: the handling of time and the level of aggregation of elements or level of details. If time is continuous, then it is not reasonable to look at single elements in the system, because they disturb this point of view. If time is not continuous, then it is not reasonable to look at elements in an aggregated manner, as it is then difficult to set point in times where the elements change their state.

3 THEORETICAL FOUNDATIONS AND MODEL FRAMEWORK

DES models are characterised generally by a greater level of detail than SD models. Time is not continuous in DES models, it is chopped by events. An event is a point in time at which the state of the system changes. So only the occurrence of events is of interest, not so much the time in between; there is usually a function that keeps track of the time in DES models and it jumps from event to event.

(Zeigler, 1976) introduced the Discrete Event System Specification (DEVS), which forms the theoretical basis for most DES models:

$$DEVS = (\chi, S, s_0, \Upsilon, \delta, \lambda, \tau)$$

DISCRETE
EVENT
SYSTEM
SPECIFICATION

where

χ is the set of input values;

S is the set of partial states;

s_0 is the initial partial state at t_0 ;

Υ is the set of output values;

δ is the transition function with $\delta : Q \times (\chi \cup \{\emptyset\}) \rightarrow S$, where

$$Q = \{s, e \mid s \in S, 0 \leq e \leq \tau(s)\}$$

is the state of the system with e representing the time that has elapsed since the last transition of the system, $q_0 = \{s_0, 0\}$ as the initial state (like in the DTSS) and \emptyset is the absence of values;

λ is the partial output function with $\lambda : S \rightarrow \Upsilon$ as an auxiliary for the full output function Λ with $\Lambda : Q \rightarrow \Upsilon$, defined by

$$\Lambda(s, e) = \begin{cases} \lambda(s) & \text{if } e = \tau(s) \\ \emptyset & \text{if } e < \tau(s) \end{cases}$$

and

3 THEORETICAL FOUNDATIONS AND MODEL FRAMEWORK

τ is the time advance function with $\tau : S \rightarrow \mathfrak{R}_0^+$.

The state transition function δ is further split up into an internal δ_{int} and an external δ_{ext} one with $\delta_{int} : S \rightarrow S$ and $\delta_{ext} : Q \times \chi \rightarrow S$. The time advance function is only proceeding if an internal event is taking place; this is also called the transitory state ($\tau(s) < \infty$). If no internal event is happening, then the time does not advance, no output is produced; this is also called the passive state ($\tau(s) = \infty$). There is only one ill-defined event: if an internal and an external event occur at the same time. This is handled differently by every DES software. The simulation time is thus discontinuous and only calculated in the case of an internal event.

While discontinuous events can be effectively modelled with this formalism, continuous relationships with dynamic feedback cannot be shown. In that sense the quote of (Maine and Iliff, 1985) is applicable, that “the theory of parameter identification for continuous-time systems with discrete observations is virtually identical to the theory for discrete-time systems in spite of the superficial differences in the system equation forms”, but only as long as no major feedback rules are active within this system.

However, it is possible to model discontinuous events within a DTSS formalism if one accepts the calculation errors that occur if an event falls in between two timesteps and so it is possible to embed discrete events into a SD framework.

Figure 8 shows the realisation and the calculation error that is made while employing a DTSS formalism to discrete events. The vertical arrows denote those discrete events. They might fall in between two timesteps Δt and then the new state is only calculated after the next elapsed Δt , so there is a deviance $< 1\Delta t$ of a DTSS compared to a DEVS in calculating one discrete event exactly. This calculation error can be minimised by increasing the time granularity, which means lowering the distance between two Δt . However, this increases the computational power needed to calculate the state space. A DEVS-based simulation tool will only calculate the new state of the system when an event occurs and can save computational power compared to a DTSS-simulation.

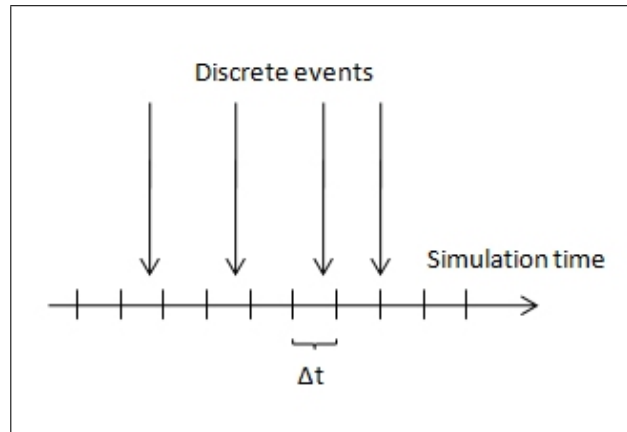


Fig. 8. Discrete events on a continuous time scale. Vertical arrows are discrete events that fall on a continuous time scale, the horizontal arrow. This arrow is divided by small time-steps Δt .

The above mentioned discussion can be seen as an indicator for the differences in the world views of SD and DES: the DES community is much more concerned about the accuracy of the model than the SD community. This is closely related to the perception of time: if time is continuous, there might not be the need for picturing all elements separately. But if time is discontinuous, looking at single entities is much more favoured.

There are two extreme points on a continuum of natural phenomena: either to look at them on an accumulated basis or to observe the behaviour of each single item. The question which quantities can be viewed as an entity is based on the question whether *a)* the processes are influencing existing model variables in a similar way, *b)* the processes' behaviour is similar in terms of modelling outcome and *c)* the aggregation is not harmful to the use of the model (Alfeld and Graham, 1976).

SD is usually taking the point of view that the structural relationships are determined by the interactions of aggregate entities. (Chahal and Eldabi, 2008)

3 THEORETICAL FOUNDATIONS AND MODEL FRAMEWORK

remarks that the accumulated view supports a strategic thinking, which is also reflected in the literature, representing the use of SD models. In general, it is predominantly believed that there is a distinction between detail complexity and dynamic complexity (Senge, 1990) and that the latter one is both much harder to understand and has more impact or leverage on a system's performance than the former one.

In contrast to that there is also the opinion that micro-behaviour is a main determinant of a system and an accumulated treatment may yield false results (Axtell, 2010). DES in contrary sets its focus on single items in a system. Those items represent the state of the system and they only change if an event happens and thus are DES systems measured with integers (which might be an advantage in simulating non-dividable entities).

The minimum level of detail for every system cannot be determined as a general rule. Furthermore, though often used there is no unique definition of what constitutes a "complex system". This topic has been examined in various ways with different focus, but one definition could be agreed by most of the authors: a complex system is a system where the parts of it interact in a non simple way (Simon, 1962). In almost all cases some kind of hierarchy is the foundation of analysis and also decomposability of the system.

While the non-trivial interaction of the constituting parts may be the dominant definition of complexity, there is no agreement whether the observed behavioural patterns of the system arise out of the interactions between the single objects (Axtell, 2010) or as a result of some structural characteristics (Lane, 2000).¹⁶ This might also be a question of the observed time span. While an analysis of short time spans may require a thorough observation of the complexity that arises out of details (Senge, 1990), the analysis of system's behaviour over a longer time span

¹⁶Complexity for computational solvable problems is set by the problem of computational power: whether or not problems could be classified as non-polynomial, i.e. more formally if the assumption $P = NP$ does not hold. This has serious implications on optimisation theory, because it touches the question whether or not it is possible to find best or most efficient algorithms for some distinct kind of problems.

3 THEORETICAL FOUNDATIONS AND MODEL FRAMEWORK

may be the result of the interaction of instances that are higher up in the hierarchy of the system.

Besides the notion of time and the world view concerning the level of detail in a model there is a further distinguishing factor between SD and DES: the handling of stochastic elements. This can be best illustrated by the categorisation of simulation models as made by (Law and Kelton, 1991):

1. static versus dynamic simulation models,
2. deterministic versus stochastic simulation models,
3. continuous versus discrete simulation models.

Though general statements are often problematic, most SD and DES models can nevertheless be allocated to the categories in each of the three axes as shown in table 1.

Tab. 1. Allocation of SD and DES to simulation model categorisation

	SD	DES
Axis one	dynamic	dynamic
Axis two	deterministic	stochastic
Axis three	continuous	discrete

This categorisation is not a software restriction; however, it is not by chance that the adjustments in the different software favour the use of either a deterministic or a stochastic handling. There are multiple purposes of a simulation model: prediction, performance, training, entertainment, education, proof and discovery (Axelrod, 2006). The SD community has clear objections on the possible application of their models for the purpose of predictions (Sterman, 1991). It is often not necessary to apply stochastic elements, because SD models do not aim for point predictions, rather for tendencies within a system. These tendencies should not change much if stochastic elements are introduced; otherwise robust policies

trying to influence the system's behaviour are less probable. The DES community, on the other hand, finds it often mandatory to include randomness as a constituting element within their models (Morecroft and Robinson, 2006). That is due to the fact that in their beliefs that factors outside the influence of the stakeholders of the model exert significant changes within the system and these influences are best captured by randomness. So in the DES worldview a deterministic model would obscure the "real" behaviour of a system.

3.4 Simulation models in the assembly line and semiconductor context

Simulation models have been extensively applied to analyse and optimise the behaviour of assembly lines. A reason is clearly that even very simplified problems associated with assembly lines such as simple line balancing, resource allocation or bottleneck detection are truly NP-hard (non-deterministic polynomial-time hard).¹⁷ Thus there are no general algorithms that work for each subclass of problems equally and predictably well. Also it is not clear whether an optimal solution to those problems exists at all.

Most of the problems associated with assembly lines reported in the literature are optimisation problems. In the line balancing case¹⁸ an optimal distribution of product flows is desired such that idle capacity is minimised respectively capacity usage is maximised for lowering the costs per unit. A similar case is the resource allocation where given resources are maximised. That is usually applied in cases where there are less operators than machines. One big research field, especially linked to the semiconductor industry is the determination of an optimal scheduling plan. This, too, is a maximisation problem where the overall output is increased by given resources. Only bottleneck detection can not being considered an optimisation problem.

¹⁷See (Goldwasser and Motwani, 1999) for a definition of NP-hard problems and the application in assembly line-related problems.

¹⁸According to the review of publications in the field by (Mahayuddin and Tjahjono, 2010) the most discussed issue in the field.

3 THEORETICAL FOUNDATIONS AND MODEL FRAMEWORK

However, optimisation may be problematic for the following reasons:

- it is difficult to define the objective function. There are quite often divergent goals, and defining what should be optimised is then a trade-off;
- the restrictions are only in some cases linear functions. Mathematical programming methods do have strong difficulties dealing with non-linearities;
- the robustness is not always given. External uncertainties and initial conditions can exert strong influences that undermine the founded optimised solution.

Simulation models can partly meet those objections as multiple scenarios can easily be tested. It is well recognised that there is no unique simulation approach towards best tackling assembly line problems (Owen et al., 2010). However, it has been claimed that it is not possible for SD for dealing with the high detail complexity of assembly lines (Godding et al., 2003) and that DES is the one simulation method best suited for it.

SD studies focus more on modelling the supply chain integration (Rabelo et al., 2004; Speller et al., 2007; Baines and Harrison, 1999). (Godding et al., 2003) state that SD seems to be inappropriate to model the assembly line due to their lack of “granularity needed to model the complex stochastic material flows”. On the other hand (Filho and Uzsoy, 2010) postulate that “manufacturing system modeling represent a missed opportunity for SD modeling”. This study wants to make the first step into that direction.

The main reason why the semiconductor industry has gained so much attention in simulation modelling¹⁹ is the high capital costs in acquiring and maintaining plant equipment, combined with a low production volume per product type, short

¹⁹ (Semini et al., 2006) present in their literature review the number of applications by industry of simulation studies, published in the proceedings of the annual Winter Simulation Conference. The semiconductor industry is the industry with the highest number of applications.

3 THEORETICAL FOUNDATIONS AND MODEL FRAMEWORK

lifetime cycles and the multitude of production sequences.²⁰ As some of the processes occur more than once in the whole manufacturing process it is thus possible and economically indicated to use the same machine twice. Therefore no classical assembly line like in the automobile producing industry exists where each product is flowing only in one direction and does not have circularities in its manufacturing process. This adds further difficulties for both designing a simulation model and deriving policy recommendations. Indeed, it has been claimed that the semiconductor manufacturing process is the most sophisticated production process in the history of industrial production (Mason and Fowler, 2001). For an overview of the production processes refer to (May and Spanos, 2006).²¹

Simulation projects with SD in the semiconductor industry are mainly models of supply chains. (Bezemer, 2003) describes SCM policies based upon a model developed by group model building sessions in order to reduce lead times. (Filho and Uzsoy, 2008) examine how cycle times could be reduced with defining optimal lot sizes, the study of (Gonçalves, 2003) focuses on the analysis of phantom orders and their effect on capacity utilisation and (Minnich and Maier, 2007) describe the advantages of a pull- distribution system compared to a push-distribution. However, SD seems to be underrepresented in semiconductor manufacturing modelling. That might be due to the fact that it has been shown that “simple” models could be misleading in such a complex manufacturing environment (Rose, 2000) and SD modellers avoid building models of that detail complexity.

However, with the aforementioned DTSS-formalism it is possible to include single entities and by this it is possible to go from an abstract level, at which most published SD models are settled, to a concrete, practical case, which is essential for a successful implementation of a simulation study in practise (Coyle, 1975). The problems of the theoretical framework for a multi formalism approach, combining continuous and discrete events simulation paradigm as done by (Zeigler et al.,

²⁰This need not to be the case for every manufacturer in the semiconductor industry; however, (Johnzén, 2009) states that in most cases these are indeed issues and that a worsening of the situation is expected.

²¹They are characterized by a multitude of single operations or sequences, with some repeated steps at various stages in the production process.

2000) can be avoided; and there is, besides that theoretical work, hardly any applied literature on this topic (Vangheluwe et al., 2002). Multi-formalism have only been applied in practice in hybrid modelling as discussed in the literature review. Only (Dvergsdal, 2006; Linge, 2007) have used a similar concept within a SD framework; they were exploring the patient flow through an emergency department, respectively through a surgery unit, at a university hospital in Bergen, Norway.

A System Dynamics model, if well designed and approved by the stakeholders of that model, could help to answer two fundamental questions, both which are interrelated. The model states a relationship, saying that x causes y . Assumed that y is the desired output the real system should deliver, then the model can answer the questions why one should do x , first of all, and why one should value x , second of all (Vázquez and Liz, 2007). Valuing x means also accepting counter-intuitive behaviour in order to achieve the desired result.

This study tries to demonstrate that no hybrid or high-level-architecture models (Venkateswaran and Son, 2005) are necessary to include discrete events in a SD model and to successfully model detailed features of an assembly line. The framework of SD can provide a standard diagramming method; something that has not been implemented in the DES community so far (Morecroft and Robinson, 2006). In a way it may aim at contributing a small step towards the “discovery of which aspects can be meaningfully »studied in isolation for the sake of their own consistency«, in other words: with the discovery of useful and helpful concepts.” (Dijkstra, 1982)

4 The simulation model of the laser production

In order to come to satisfactory information integration the initial focus of knowledge elicitation was the establishment of Group Model Building (GMB) workshops, as suggested and practised for example by (Bezemer, 2003). GMB is helpful in involving the problem owners into the model building process and it is a technique that is suited especially for the development of SD models. This is supposed to develop confidence in the model by the internal clients, as represented by senior

line managers, capacity planners and experts of supply chain integration (Vennix et al., 1996).

However, the limits of this method were already discovered in the very first session, which was mainly focused on defining the reference mode (Keating, 1998) and some important causal relationships. Here, though a general agreement on the model purpose could be achieved, the complexity of the issue seemed to be too big to follow a normal GMB procedure. Modelling the influence factors as described in precedent chapter requires a very detailed model of the assembly line. This is a highly technical task but it exactly meets the client's request.

Furthermore, it was not possible to assign different roles to different persons: in GMB normally a moderator, one or two facilitators and modellers is standard. The difficulty to fulfil all those roles with one person was too high (Andersen, 2010) for sticking to the GMB method. Supposedly the difficulties in fulfilling different roles hinders personal congruency: as neutral facility or communication is not possible (Visser, 2007), limiting the scope of the work on the one hand and moderating a creative group process on the other hand is physically not possible for one person.

There is also another aspect of refraining from choosing GMB: simulation itself has besides some small projects not been used at Osram and also the knowledge of SD was not widespread. Introducing a new simulation and problem structuring method in combination with a new group process and the extension of classical SD methods was maybe a bit too ambitious. The experience that establishing new group processes in an industrial setting are difficult seems to be quite typical for industrial simulation projects, where mainly the result is of interest and not so much the detailed configuration of the model (at least in those cases where the models will probably no being reused) (Brandolini et al., 2008).

So a typical “expert model”²² seemed to be a reasonable and feasible approach.

²²A model whose equations are only understood by a limited range of stakeholders. This is usually the case for DES models, which are usually not used e.g. as training tools and are more the domain of simulation experts (Sweetser, 1999).

4 THE SIMULATION MODEL OF THE LASER PRODUCTION

The coding procedure for the model was substantially higher than for a typical SD model, confirming the observations of (Tako and Robinson, 2008) who compared the model coding procedure times of a DES modeller to a SD modeller (the concept of a DES model is closer to this model than the “traditional” SD concept).

The model could have been implemented both in a SD or a DES software. However, SD was chosen because it opens up for a broader range of analysis. With SD it is possible to include feedback processes on the load of the assembly line, on the staffing of the different production processes or on long-term capacity planning. SD lays the basis for a longer-term assessment of cost-benefit analysis and the transition of including more parts to model a more holistic perspective of the system is obvious.

The pitfalls of increasing the level of details (in comparison to a traditional SD model) are described by (Chwif et al., 2000), who claim that with a rising level of detail the confidence in the model decreases. The three reasons for increasing complexity mentioned, *i*) the “show off” factor, *ii*) the “include all” syndrome and *iii*) the “possibility” factor, are not the reasons for the complexity of this model; the complexity needed to accurately mimic the behaviour of the assembly line is inherent in the semiconductor manufacturing process. Indeed, more complexity would have been desirable at some point in order to demonstrate the loss factors in capacity that arise by an insufficient level of qualification of part of the operators.

On the other hand technically sound or well designed models are not helpful if the people within the organisation who have the power to take actions to change the current situation don’t trust them. Only if the model is accepted by its stakeholders there is a chance that also the simulation results are accepted. Here SD has an advantage over DES models because of its standard diagramming method which can be explained quite easily. This helped in confirming the different production paths of the chosen product types.

The acceptance of the model is a “necessary, but not sufficient condition” for the acceptance of the simulation results (Morecroft, 1984). He describes the inter-

action and feedbacks from simulation results to a debate and discussion on strategy that lead back to a better understanding of the system, which may drive some further simulations with different assumptions. The acceptance was gained through several group discussions in so-called “validation workshops”.

A last point may serve as a reason for choosing SD as a problem structuring method: prediction was not a main purpose of the simulation. It is a widely held belief in the SD community that prediction in a classical sense should not be the goal of a SD model (Stermann, 1991). Instead, there are several other reasons for a simulation project besides predicting an outcome, according to (Epstein, 2008):

- explain the structure of the system,
- illuminate core dynamics,
- discover new questions,
- illuminate core uncertainties and
- demonstrate trade-offs.

SD can meet those reasons. It helps to clarify a messy situation and is able to give a quantification of different policies. In making the structure of a system concrete and laying the foundations of discussion with the graphical representation of the system SD helps to reveal contradictory assertions. These assertions come from the mental models of the different stakeholders. The following quote from Dijkstra, made in the context of developing a computer programme, is also valid in a management’s context:

If your specifications are contradictory, life is very easy, for then you know that no program will satisfy them, so, make “no program”; if your specifications are ambiguous, the greater the ambiguity, the easier the specifications are to satisfy (if the specifications are absolutely ambiguous, every program will satisfy them!). (Dijkstra, 1982)

It may seem superfluous to ask for goal consistency; however, in daily life operations it becomes too often reality that decisions are made upon recent requests or observations and that those requests may not serve an overall or stringent purpose. One can easily get lost in details; but a management's task is primarily to keep track of the "big picture". SD is an appropriate tool for setting the focus on long-term goals, deepening the understanding of the system and measuring impacts of goal-supporting or -contradicting policies.

4.1 Description of the model

The model represents the processes in the end-of-line part assembly line of the laser production at Osram, Regensburg. The production is a batch production; however, the lot sizes are not the same throughout all processes.

There are a multitude of different products running on that assembly line. Not all products use the same resources as some machines are only suited or necessary for certain products. Some processes are purely manual. The laser production has been described in section 2.1; the process flow chart (figure 2 on page 6) gives a comprehensive overview of the resources included in the model as well as an impression of the complexity of the product flows.²³

The production is a batch production due to the small turnover contribution by each product type as the total demand for each product class is too small compared to the high costs for machine acquisition to justify the establishment of an assembly line for every single product. Also for profitability reasons the production department has to share its production line with the product development department, hindering the production to be fully effective, but enabling ramp-ups of new products to be fast and effective.

The model was built with System Dynamics software - Powersim Studio 8, service release 5a, from Powersim Software AS. Even though the structure is too large to discuss it here comprehensively, some assumptions for simplifying the

²³The flow of products is shown in appendix A on page 94.

4 THE SIMULATION MODEL OF THE LASER PRODUCTION

model were necessary. Those simplification are:

- (a) only five products have been considered; in reality these five products represent only around 70-80% of the total production volume
- (b) only products sold commercially are considered in the model; no prototypes have been implemented into the model as the processing times vary much and cannot be predicted
- (c) the times for manual processes are averaged; they are taken from current reporting and calculation systems, but average values give rise to imprecision. Statistical deviances in the manual processes are thus not considered
- (d) the times for machine processes are averaged. The reported value of the outcome is jigs, but the yield is not considered. The lot size does not vary in the simulation; yield losses could lead to a reduction in processing times
- (e) not all processes are modelled in exact detail; some of the processes are combined as they are processed sequentially; the small steps (with machine idle times) in between are not considered
- (f) stops in the production process are not considered. Usually, lots are stopped for different reasons like unexpected process deviations
- (g) the manning of the shifts are not incorporated into the model. Some shifts could have less operators due to holiday or sickness absence
- (h) the qualifications of the operators is not considered; there is no distinction within each cluster and between different shifts²⁴
- (i) machine downtimes are not considered

The reason for drawing those assumptions is not practical feasibility, rather interpretability. Testing the influences of scheduling, prioritisation and bottleneck removal for themselves is challenging enough due to the high number of possibilities

²⁴Since for some processing more than one qualified operator must be available at the same time, unbalanced operator availability or an unbalanced qualification level may lead to a stop of some essential processes or even the whole assembly line.

arising out of those influence factors. Including some of the variables mentioned above would have obscured the result unnecessarily. All those simplifying assumptions were discussed with the stakeholders of the model.

Appendix A on page 94 gives a model documentation; the process flows of the five products are shown as well as some particularities regarding lot sizes and conflicts in between processes.

4.2 Modelling principle

There are five product types incorporated into the model with different process flows or requirements. This is realised in the model using an array-structure for the products. The products themselves are modelled as discrete entities.

For realising a model that keeps track of single product entities it is mandatory to include the timestep in the modelling functions. The theoretical foundation for this (the DTSS formalism) has been discussed in chapter three.

The operators are divided into four clusters and there is not one operator for each single process available. They are distributed by an algorithm in Powersim: `priorityallocdiscrete`. So the operators could be a source of scarcity or form the bottleneck of production as well and not only the machines. Not all machines are incorporated in detail into the model; certain processes justified an aggregation.

Figure 9 shows the modelling principle schematically. In this section the flow of products as shown in figure 9 is described in more detail.

Consider a product x_i (where the index i denotes the number in the aforementioned product array) that enters the production process at time t_1 . The product array a_j would thus look like

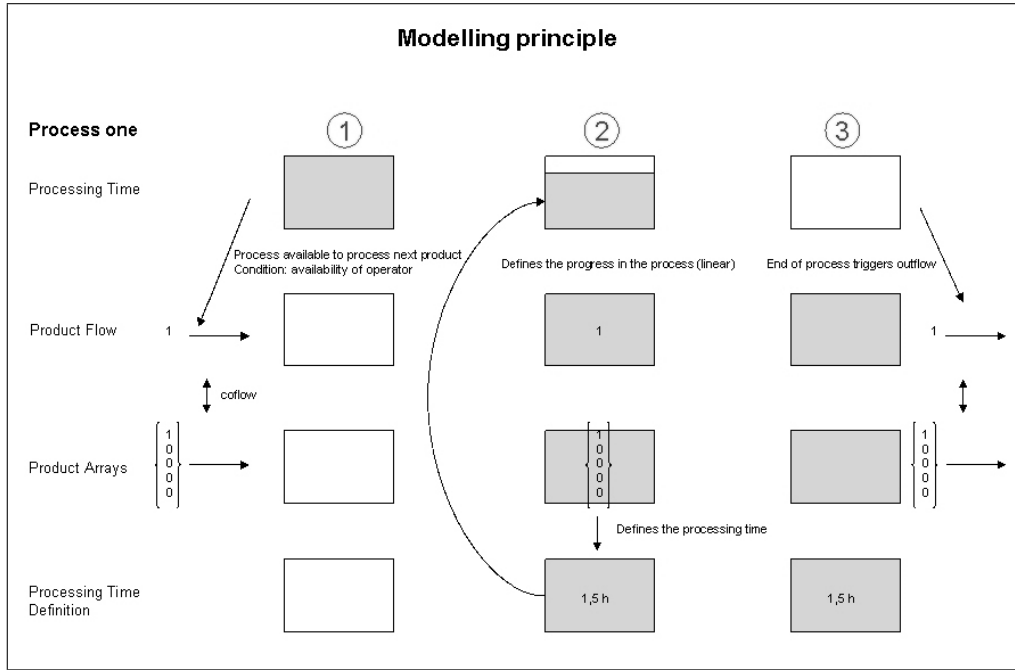


Fig. 9. Modelling principle

$$a_j := \begin{pmatrix} 0 \\ \vdots \\ x_i \\ \vdots \\ 0 \end{pmatrix}.$$

The array a_j is introduced into the line via an ordinary pulse-function. It is transferred to a stock s . This stock represents one manufacturing process in the line (compare that also to figure 2 on page 6). Stock s is defined at time t as

$$s_t := \begin{cases} \frac{a_{j(t-1)}}{\Delta t}, & \text{if } \pi_{t-1} = 0, \\ a_{j(t-1)}, & \text{otherwise,} \end{cases}$$

with π as an auxillary stock, keeping track of the elapsed processing time for product x_i . The initial value for π is 1.

The definition for π in continuous notation is

$$\pi_t := \pi_0 + \int_0^t (\sum \text{flows}) dt,$$

which is approximated in the simulation by

$$\pi_{t+\Delta t} := \pi_t + \Delta t * \sum (I_t - O_t)$$

where I represents the inflow to π :

$$I_t := \begin{cases} \frac{1}{\Delta t}, & \text{if } \pi_{t-1} = 0, \\ 0, & \text{otherwise,} \end{cases}$$

and O represents the outflow of π :

$$O_t := \begin{cases} \frac{\pi_{t-1}}{\Delta t}, & \text{if } \frac{1}{t_p} * \Delta t > \pi_{t-1}, \\ \frac{1}{t_p}, & \text{otherwise.} \end{cases}$$

t_p is the process time for product x_i , and it is also an i -dimensional array with the process times for every x_i for process s . So when a_j enters s , then t_p is multiplied with a_j .

The process time t_p is further divided into a machine and an operator time, so π is in fact a stock with a two-dimensional array (or matrix). The interpretation and calculation is straightforward. The array-structure has also another advantage: with it it is possible to assign each product for its ascertained process. This is also done by multiplying the product matrix a_j with the production time, so t_p , the production time for x_i is set to 0. Note that it is possible to introduce process deviations into the formula: the first case for I_t has therefore to be changed from $\frac{1}{\Delta t}$ to $\frac{\xi(t)}{\Delta t}$ with $\xi(\cdot)$ as the “white noise” (Sterman, 2000) of the system at time t .²⁵

Figure 10 shows the full realisation of one building block of one process. The equations of this basic building block can be found in appendix B on page 96.

²⁵Though randomness could be incorporated into the model this was not desired for simplification purposes.

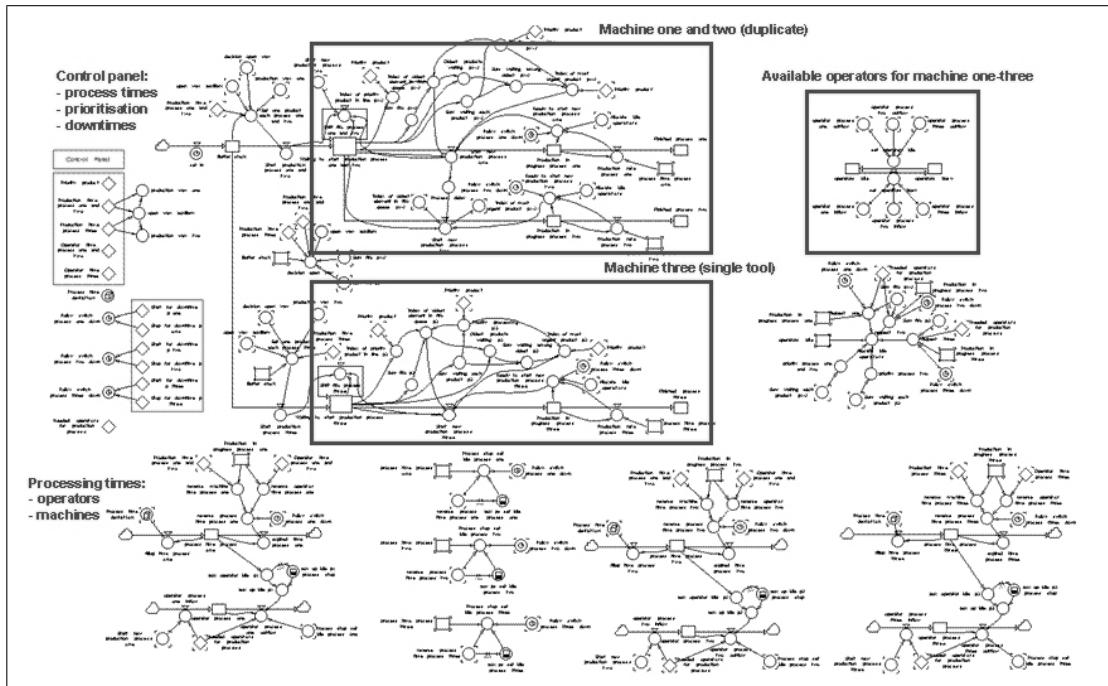


Fig. 10. Basic building block of one process

Though some of those formulas are peculiar equations from Powersim, the fifo-rule can be implemented in any SD software that is able to handle matrix structures.

One problem is so far unsolved: the inflow for s is normally a_j and it comes, if not from an external inflow as for process 1 and 2, from an intermediate stock b . However, b does not keep track of the order of the incoming x_i in a simple, i -dimensional array structure, because it will only increase the number for x_i . For resolving this problem a matrix structure for b is introduced.

Product array a_j enters the fifo-stock b . This stock is a matrix structure that looks like

$$b = \begin{Bmatrix} a_{jn} & \cdots & a_{jm} \\ \vdots & \ddots & \vdots \\ a_{kn} & \cdots & a_{km} \end{Bmatrix},$$

and each time a new product array enters the stock the matrix performs an

4 THE SIMULATION MODEL OF THE LASER PRODUCTION

internal shift for each column n, \dots, m :

$$(a_{jn-1} \rightarrow a_{jn}) \quad \forall j, \dots, k.$$

If the process in stock s is ready for taking up the next product (which means that there is no production in process and a sufficient number of operators available), then $a_{j \max(n)}$ is selected for the inflow for s . The matrix structure is for the flow again reduced to the i -dimensional array structure. The dimension of the matrix is defining the size for the fifo-stock b by the number m . Every intermediate storage facility has a maximum capacity of items to handle. So when the capacity limit is reached, the model produces a calculation error by collecting all a_j at column m .

In Powersim the transpose-function augments the product-array to a matrix, the inverse is the collect-function. The shifting is done with the help of a function that fills up all non-defined items in the matrix with zeroes. Appendix B gives the details of the realised structure. The integration order for the fifo-stock b is zero. This is an advanced modus in Powersim, allowing the flows being discrete and so not depending on the length of the timestep Δt or the integration method (which can by this method be higher than 1st-order Euler, for example 4th-order Runge-Kutta).

The fifo-rule does only apply if there is no priority lot waiting in b . In that case the prioritised product is selected. The weight of the allocation of the operators to the different jobs by the priorityallocdiscrete-algorithm respects prioritisation of a product as well. A deeper discussion of the prioritisation at Osram can be found in the next chapter.

With the presented structure it is possible to model process flows with SD software, something that was only thought being possible with a DES or an Agent-Based model.

However, note the inherent error in the used DTSS formalism compared to a

DEVS formalism: this error is given by the formula for the outflow of π , I . If the production time t_p falls in between two timesteps Δt , there is an error that is smaller than one Δt . The realisation of the fifo-rule needs two buffer stocks: the fifo-stock b and one further auxiliary stock that sorts the incoming flow: if $x_i > 1$, then more entities than physically possible would be processed. The auxiliary stock divides all a_j with $x_i \geq 2$ into single entities, which are processed one after each other. So the actual error between two processes (or, more formally, between s_1 and s_2) is then at maximum $< 3 * \Delta t$. Compare this also to figure 8 on page 31 and the related discussion to it. A reduction of this error is achieved by choosing a small Δt , bearing in mind the trade-off between high accuracy and required computational power. The procedure for choosing Δt in this model is discussed in the following section.

4.3 Validation of the model

The model of the assembly line was developed in close collaboration with several departments involved in production planning, monitoring and steering, product development and logistics at Osram. The assumptions mentioned in section above were checked several times by members of the different departments. The usefulness of a model can only be judged by its stakeholders (Sterman, 2002), and to guarantee that, several workshops and interviews were hold. The boundaries were found to be adequate and the model does not violate any physical laws. So, from a stakeholder's point of view the model can be called sound.²⁶

Various tests have been performed with the present model. They are listed in table 2. For a detailed description of those tests refer to (Sterman, 2000). Of course, validation of a model is never possible, as all models are a simplification of reality and thus wrong (Sterman, 2002), but they help to increase the trust in it. However, some of the tests are indispensable because they aim for mathematical

²⁶Some of the assumptions have been discussed and they were criticised by some of the stakeholders. However, for enabling a feasible analysis of the results (which means that there are not too many variables in the model) the version of the model that is the basis for this analysis could generally be accepted.

correctness of the model.

Tab. 2. Overview over the performed validation tests

No.	Name of the performed test
I	Structure assessment test
II	Boundary adequacy test
III	Extreme conditions test
IV	Dimensional consistency test
V	Integration error test
VI	Behaviour reproduction test
VII	Parameter sensitivity test

One main message of the SD community is that structure defines behaviour (Schwaninger and Hamann, 2005). In that respect it is absolutely important that the structure of a model is approved by its stakeholders. This was achieved here by multiple discussions with members of different departments and a final agreement was reached. This is a very crucial step of the modelling process. SD modellers usually do not perform a parameter fitting procedure, so there is no ϵ as a deviance parameter. SD is here different to statistical methods like structural equation modelling, where the latent variable η is explained by some indicators ξ_1, ξ_2, \dots , which are derived by a set of items $\delta_1, \delta_2, \dots$. Though those models may also explain multiple influence parameters on the variable of interest it is not possible to insert feedback loops as in SD models. In statistical models there is a strong focus on the fit of the model, while in SD the focus is more on the structural relationship. By confirming its structure by the stakeholders a SD model passes the structure assessment test (I). The boundaries of the model were defined at the very beginning of this study and they are, by approval of the stakeholders, adequate (II).

No errors in the coding remain in the model. Each subsystem of process modules (compare to figure 10 on page 46) was tested separately and in combination. Extreme conditions like the absence of products or operators, the downtimes of crucial processes or overloading the system with too many products were tested

4 THE SIMULATION MODEL OF THE LASER PRODUCTION

and all those tests showed consistent behaviour (III). If necessary conditions for the production process are absent, no output is generated. All units and dimensions were found to be consistent (IV).

The model is not sensitive to the integration method itself; so it makes no difference whether Euler or Runge-Kutta is used and what order of integration is chosen (V). However, the granularity of the timestep does make a difference. The reason for it was already explained in the section above. Finding an adequate solution for the trade-off between accuracy of the simulation and the time request for performing the simulation several simulations with increasing timestep were performed until first deviances became prominent. The biggest timestep without deviances compared to a very small one was then chosen. The maximum deviance possible in the simulation with the chosen timestep is smaller than 2%.²⁷

As a model is a simplified picture of reality there will naturally be some deviance of the simulation results from actual system's behaviour. The strength of a SD model is that it "has to be right for the right reasons" (Oliva, 2003), meaning that with an iterative calibration test one builds up confidence that no important parameter was omitted. A validation against an actual production week was performed (delivery week 17 of fiscal year 2011); and for this specific week all items in the line actually being processed or waiting for being processed as well as their order at the beginning of the week were translated into the five product classes of the model. The model was initialised with those values and then the outcome of the simulation after the week was compared to the actual outcome. The number of lots was exactly the same; however, there were some deviances in the processing order of the remaining products. One reason for this may be some internal shifting or re-prioritisation which are often not recorded.

There are two reasons besides the simplifications listed in section 4.1 why the simulation results of this assembly line model does not match actual behaviour perfectly well:

²⁷This deviance is only a theoretical one; in practice it is much smaller than the mentioned 2%.

- (a) the initial conditions can have a strong influence on the simulation outcome in this case (as described below in the sensitivity analysis);
- (b) those initial conditions like the exact processing order at every stage can not be fully incorporated into the model as not every detail is reported in reality.

While most of the parameters listed in section 4.1 could have been incorporated into the model it has been chosen not to do that for keeping the interpretation of the results manageable. However, the two extra items listed above would have been deviance factors that cannot be eliminated, so the behaviour reproduction test was passed (VI).

The last test remaining is the sensitivity analysis, or parameter sensitivity test (VII). Although “[a]ll models exhibit numerical sensitivity” (Sterman, 2000), it is nevertheless important to observe behavioural change in a model with slightly changed initial conditions to test for robustness of policies. In the following the processing times at one station were altered and the results were observed over six weeks of simulation time.

For this test as well as the analysis of results in the next chapter the simulation conditions were always the same. All scenarios are standardised with the following procedure:

Initial conditions The initial conditions are defined as the status of the assembly line after one week of production from an empty line. The first week is not displayed in the results for prevention of compounding the analysis with the start of the run-up (e.g., a wrong reporting of average capacity utilisation).

Simulation time The simulation time is six weeks in every scenario. As a different product mix requires different resources and this different resource utilisation leads to a different state of the system at the end of every week, analysing only one week would lead to false conclusions. It is therefore necessary to run the simulation over a longer time horizon.

Prioritisation set For every simulation a prioritisation set is chosen and this is kept constant throughout the simulation. That said it is possible to define a

priority for product x_i e.g. for half of a week and then a priority for product x_j for the rest of the week, but this set is kept constant over the whole simulation time for each week.

Scheduling Either an all-at-once scheduling or a scheduling sequence is simulated. In the former case every week at the beginning all lots that are to be processed in that week are put into the line; in the latter case they flow gradually into the line with a constant sequence.

Shift manning The availability of operators of each cluster is defined at the beginning of the simulation. The same applies to shift pauses. Those are always at the same time for all operators at once (no alternating pauses).

A parameter sensitivity analysis should also make sense in the real system and there are some purely manual production stages within the assembly line where different processing times are probable, such as in process 3: the manual magazing. At this station the operators are putting the laser bars into coating jigs, so that different deposition methods for the mirror layers can be applied to the bar facets. As this process is driven by the individual productivity of each operator, the times underlying both this model and other calculation methods currently in place at Osram can be no more than average production times. So a reasonable assumption for a parameter sensitivity test would be to set this time up in order to simulate the effect of operators, which are newly trained and thus not as fast as more experienced operators. This assumption was tested in several scenarios.

Tab. 3. Tested scenarios in the sensitivity analysis

Scenario	I	II	III
Product 1	2	6	3
Product 2	5	4	7
Product 3	9	9	8
Product 4	8	8	10
Product 5	1	2	2
Total lots	28	35	35

4 THE SIMULATION MODEL OF THE LASER PRODUCTION

Table 3 gives an overview of the three different scenarios that have been used for the sensitivity test. Each scenario has a different product mix and total amount of lots. The basic assumptions (prioritisation and scheduling set) are kept constant. All three three product mix scenarios are run twice. Simulation scenarios with “normal” production time are called “base runs” in this analysis, where scenarios at which the times at process 3 were set up 10% are called “alternative runs”.

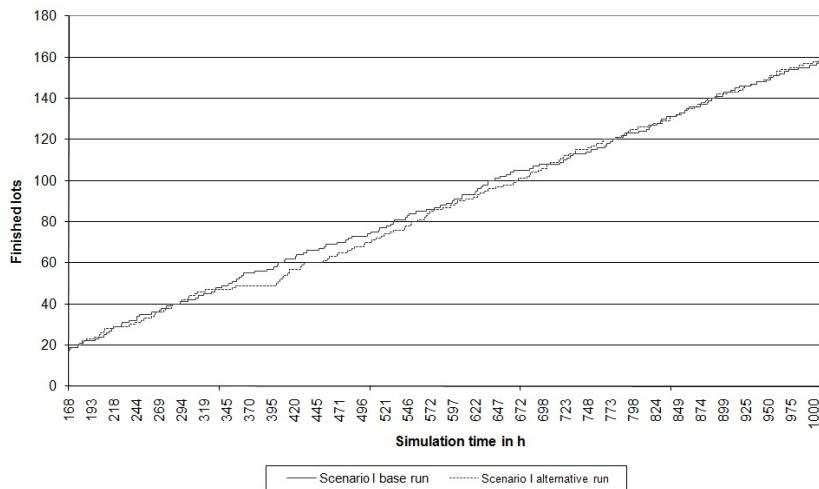


Fig. 11. Simulation results from parameter sensitivity test with scenario I

Figure 11 pictures the simulation results for scenario I, based upon the production start of 28 lots per week, which is kept constant over the whole simulation time. What is striking here is that though the alternative run has a higher production time (dashed line), the overall production output after 6 weeks of simulation is the same.

Compare the graphical result also to table 4. This table lists the completed lots per week in scenario I base run and those of the alternative run. The amount of lots introduced into the system per week is constant, so comparing the overall lots that are either work in progress or waiting before one production stage, as shown in figure 12, gives the same picture: at the end of the simulation period the work in progress is the same in both runs.

Tab. 4. Weekly production output in scenario I in lots

Simulation week	Base run	Alternative run
2	31	29
3	26	23
4	31	31
5	26	29
6	27	29

Nearly the same result as in scenario I occurs in another production setting as well: the outcome of both runs of scenario II is shown in figure 13. Here the production start per week is 35 lots.

It is remarkable in this scenario that the amount of production start is higher than the system can handle, but nevertheless the alternative run is at some stages clearly better than the base run; compare that also to figure 14, where the total products in line are pictured.

What we see here in comparing the base runs of scenario I and II (compare

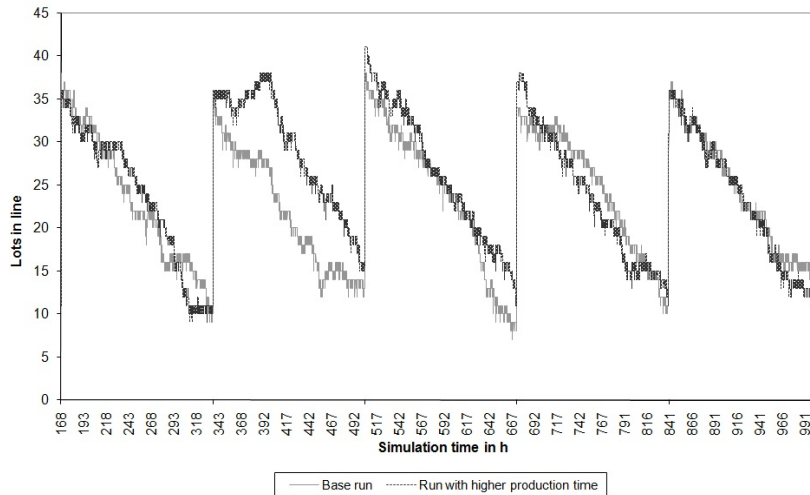


Fig. 12. Comparison of products in line of both runs in scenario I (work in progress)

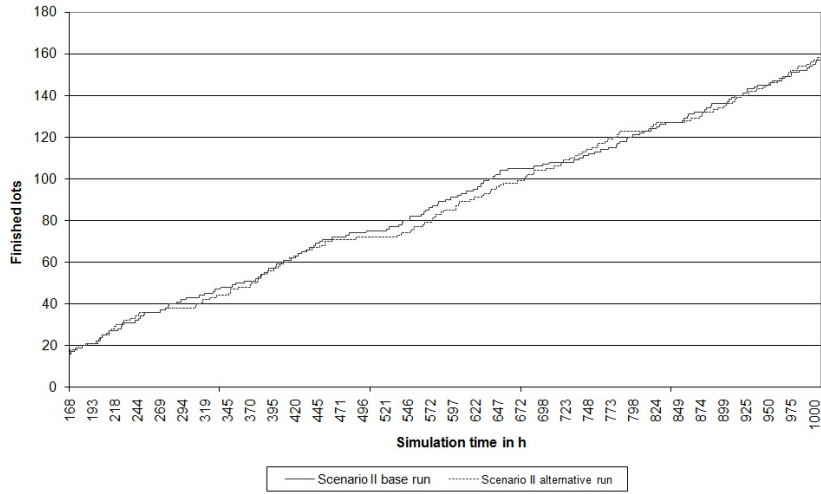


Fig. 13. Simulation results from parameter sensitivity test with scenario II

figure 11 with figure 13 and table 4 with table 5) is that the variability in the line is increasing. The output in the latter case is much more unstable and thus unpredictable. That said the frustration not only in the production department is much higher as the differences between two weeks are bigger. So in one week the rate of completion is higher than in the other week, causing wrong commitments

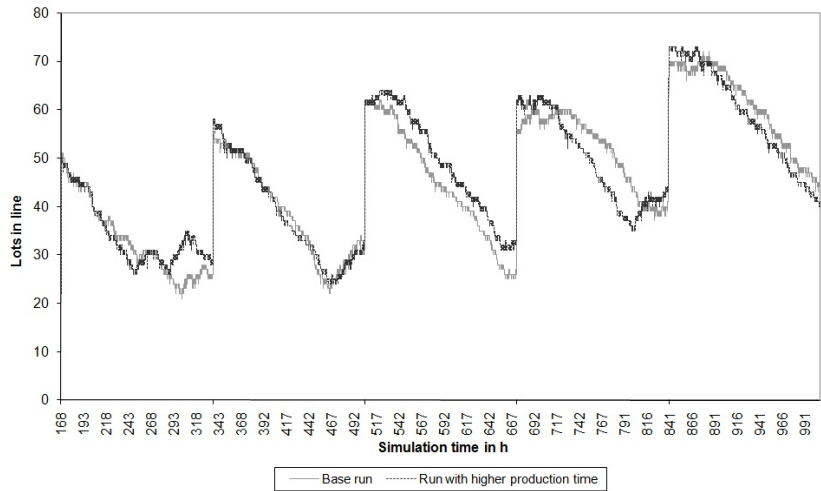


Fig. 14. Comparison of products in line in scenario II (work in progress)

to the clients, if based on one favourable week.

Tab. 5. Weekly production output in scenario II in lots

Simulation week	Base run	Alternative run
2	31	28
3	28	28
4	30	27
5	22	28
6	30	31

The overall output from scenario I and scenario II in both basic and alternative runs is almost the same. The capacity limit of the assembly line is reached, a further loading of it only increases the work in process and thus the cycle time. This is further discussed in the following chapter.

In scenario II simulation week 5 seems to be much better in the alternative run (dashed) than in the base run with 6 more lots produced. Therefore some validation runs with nearly the same initial settings as in the beginning of week 5 of the alternative run are performed, once with the normal production time and once with the higher production time at production stage 3. Only the processing orders of some lots were changed. The results are shown in figure 15.

Here the alternative run is the worst one; however, it is also astonishing that the validation run with higher production time (dashed-dotted line) is better than the underlying run from scenario II (continuous line). So even a small change in the initial settings is able to reveal this kind of deviation.

In scenario II it becomes also more obvious that despite the constant input the output follows no regularity. Figure 16 compares the completion time between two lots of the base run with the alternative run.

One can remark that there is actually no obvious pattern emerging out of this figure. A direct comparison of the completion time between two lots of both runs

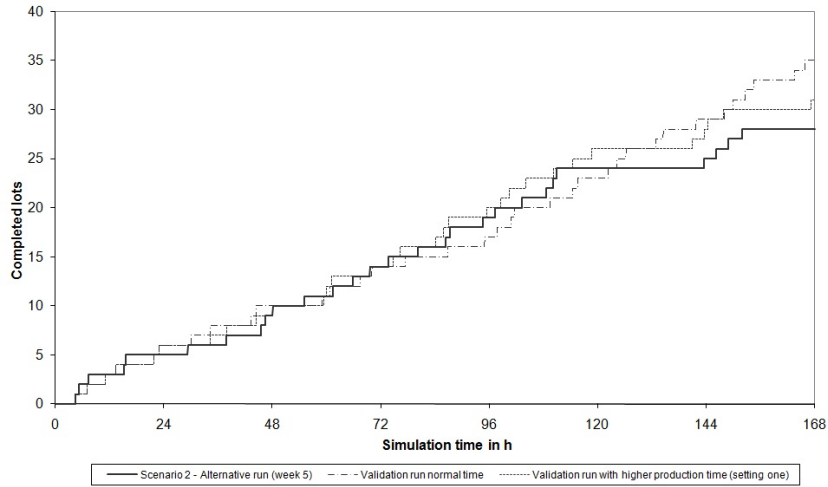


Fig. 15. Comparison between week 5 from the alternative run of scenario II with two validation runs with alternative settings.

of this scenario reveals further evidence for the counter-intuitive behaviour of the line that the run with the higher production time is not necessarily worse than the run with the normal production time. Figure 17 gives a box plot diagram of both runs (of the values that are plotted in figure 16).

These results are quite surprising; one would expect from a deterministic discrete event system with no major feedback loops a straight, predictable beha-

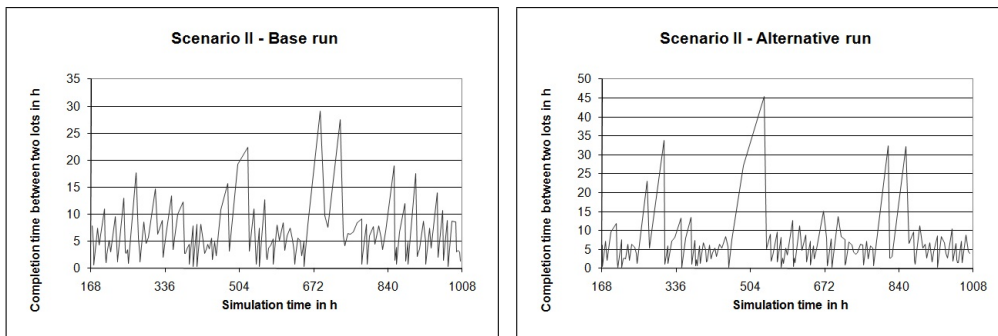


Fig. 16. Comparison of completion times between two jigs (output) of scenario II. The time on the y axis is the hours between the completion of two lots. The amplitudes are much higher in the alternative run.

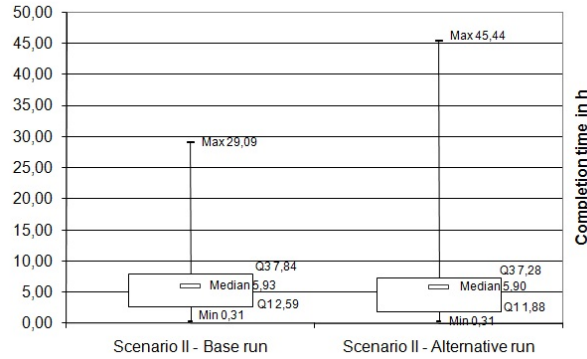


Fig. 17. Box plot of the completion time between two jigs of runs of scenario II. The values are those plotted in figure 16.

viour.²⁸ Apparently, this seems to be not the case in this system. An analytical analysis of why these results are occurring is rather difficult; however, it was found that in some particular cases a better coordination of production at later stages is obtained through this higher production time. Practical conclusions from this analysis are also discussed in the next chapter.

Scenario III on the other hand gives a more expected outcome: here the regular production start is also 35 lots per week, just as in scenario II, only with a different product type composition. But here the alternative run is not better than the base run (see figure 18). A higher production time leads in this case to a considerably lower output.

Table 6 shows that the two production lines drift apart, beginning from week 4. Week 4 and week 6 are considerably different in the alternative run. Here, a longer manual handling at station 3 sets the production outcome lower. This has to do with the different resources used by the products in scenario I and II, compared to III. In scenario III the bottleneck of production is touched by the longer production time and therefore it has an impact on overall output.

²⁸If there is the same input every week, the same output would be expected. This is not the case for any of the scenarios.

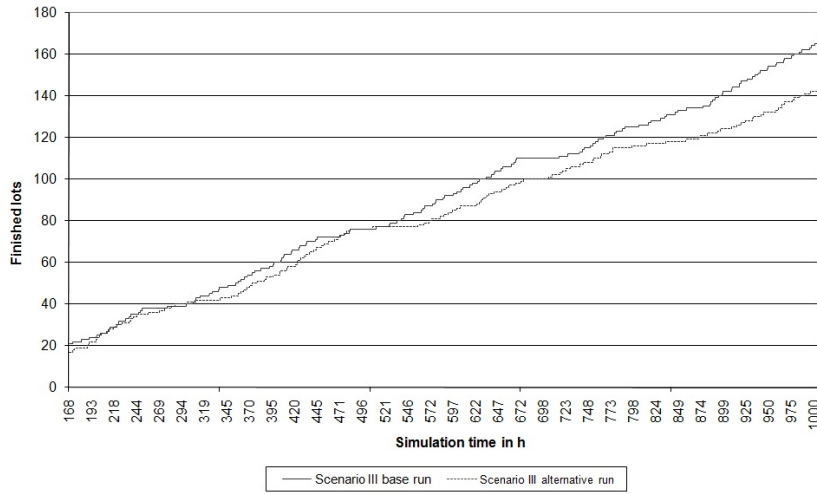


Fig. 18. Simulation results from parameter sensitivity test with scenario III

One further analysis of the last example (figure 18) may serve to illustrate that even this small deviance of parameters yield in a different outcome. In figure 19 the histograms of the base run (with the 'normal' process time) and the alternative run (with the higher process time) are compared. What have been done here is a clustering of the completion time between two lots (like the plotting in figure 16) in bin steps of two. The correlation coefficient between both histograms is 0.88, so there is a considerable deviance between both scenarios.

This simple sensitivity test already reveals some of the basic characteristics of the production line. It seems as the interactions of the products, combined with a specific production mix and the "feedback structure" of the assembly line

Tab. 6. Weekly production output in scenario III in lots

Simulation week	Base run	Alternative run
2	26	25
3	29	34
4	34	22
5	21	20
6	35	24

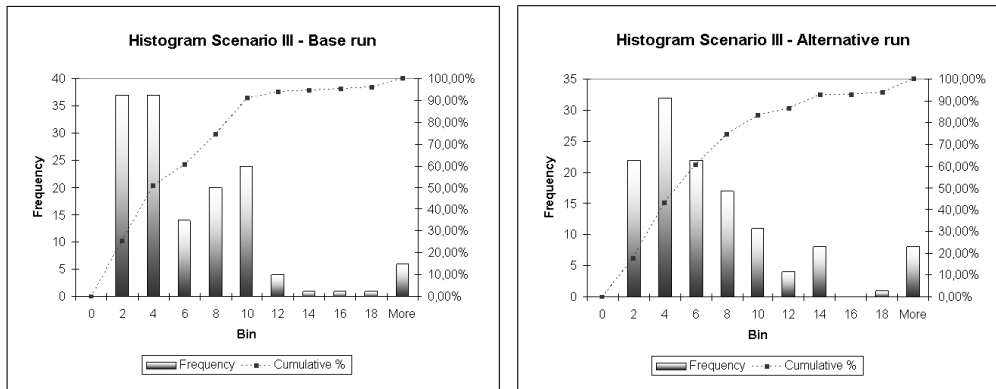


Fig. 19. Comparison of the histograms of the completion time between two lots in bin steps of two of scenario III (base run versus alternative run)

is responsible for “instability” in the line. Taking only one product as an input to the model the result is quite striking: the output is simply a straight line with a constant production rate which is determined by one process: the bottleneck.

As the research question was finding out impacts of

- prioritisation,
- scheduling and
- bottleneck removal

on line output the modelling results of those are presented in the following chapter. But the results of this analysis were quite helpful in analysing those impacts and drawing practical conclusions for the operation of the assembly line.

5 Results

Assessing the impacts of prioritisation, scheduling and bottleneck removal were guided by the question: are there any general decision rules deducible that establish a robust policy? The conditions for simulating were presented in the previous section: the same initial conditions, simulation time and shift manning is applied; for testing the prioritisation the same scheduling setting is applied and vice versa.

The main findings include:

- it cannot be judged in general whether or not a prioritisation is positive for the production output;
- it cannot be judged in general whether or not a scheduling policy is positive for the production output;
- an alignment of scheduling according to the Theory of Constraints is not deducible from the simulation results;
- the initial conditions can have huge impacts on the production output;
- in those cases where prioritisation or scheduling have a positive impact a static decision rule cannot be established; an optimal decision rule would rely on a feedback mechanism of unused (or idle) capacity²⁹;
- local decision rules like prioritisation or scheduling have a minor impact on production output in comparison to bottleneck removal, but only if the line is imbalanced;
- a balanced line give rise to problems of defining the bottleneck of production, which has some implications on the judgement of profitability;
- analytical statical optimisation cannot be used for predicting the overall production output;
- much stress on the line in terms of overloading results in oscillating behaviour.

5.1 Effect of prioritisation

The products are in general processed by a FIFO-rule, as explained in chapter 2. However, in reality there are multiple trade-off's for processing one product earlier than another product: sometimes a client insists on having a product earlier than

²⁹This can be seen as a recommendation of a pull-system (O'Callaghan, 1986).

guaranteed and sometimes there are research results of current product improvements which needs to be passed to the end of the line as early as possible. That means that following the FIFO-rule strictly would only be possible if one accepted the immediate negative effects that arise from its application.³⁰ Prioritisation is thus an important factor in the real system and that it why it is introduced into the model.

Prioritisation means that one certain product is favoured throughout the process. So if there is one prioritised product waiting in front of one process it will be processed no matter whether or not there are other products with longer waiting times as well.

Tab. 7. Selected scenarios for prioritisation analysis

Scenario	I	II
Product 1	6	1
Product 2	4	9
Product 3	9	6
Product 4	8	14
Product 5	2	2
Total lots	35	35

Table 7 gives the settings for testing different prioritisation sets. Two scenarios (I and II) should show the differences another prioritisation could exhibit. Scenario Ia and IIa have the same prioritisation: product 5 is prioritised for 50 hours, then for another 50 hours product 3. Finally product 4 is prioritised. This set is kept constant throughout the simulation. In scenario Ib and IIb product 2 is prioritised all the time.

³⁰It would be an interesting study building a SD model showing the long-term effects of prioritisation. Prioritisation works as a reinforcing mechanism up to the point where a majority of products is prioritised. Then balancing effects take over. As there are delays in this system it generates oscillating behaviour: a cyclic in- and decrease in products being prioritised.

Tab. 8. Work in process (WIP) and finished lots in scenario I with a change in prioritisation

	WIP	WIP	Finished lots	Finished lots
	<i>Scenario Ia</i>	<i>Scenario Ib</i>	<i>Scenario Ia</i>	<i>Scenario Ib</i>
Week 2	26	21	31	30
Week 3	33	26	28	29
Week 4	27	30	30	29
Week 5	41	37	22	28
Week 6	45	40	30	33

The different sets were found by trial-and-error. There are too many possibilities to successfully apply a stochastic optimisation method in order to find the “best” solution. There are 168 hours in one week and 5 different products. That means there are about $5^{168} * \Delta t = 2,7 * 10^{117} * \Delta t$ possibilities for an “optimal” solution. That said and bearing in mind that a simulation run over 6 weeks takes around 40 minutes to simulate and given that one cannot judge by simulating only two subsequent weeks (the sensitivity analysis in the previous chapter should give enough evidence for that statement) it becomes quite clear that an optimisation sequence would exceed the computing power of an ordinary personal computer by far. The first prioritisation set was deducted from an actual production week; the second prioritisation set was chosen arbitrarily.

The results of prioritisation are ambiguous. As seen in table 8 there is a positive effect of altering the prioritisation set in scenario I. The finished lots are also shown graphically in figure 20. The dashed line is scenario Ia. The output from Ia is from week 4 on always below scenario Ib.

The next example of scenario II is a counter-example of altering the prioritisation set. Product 2 is prioritised and (due to a partition after production stage 1) this product forms 18 production lots (out of 35, so more than half of the overall production!). So it could be a reasonably good idea to prioritise it: the demand is quite high and therefore the pressure to produce this product is apparently higher than in scenario I.

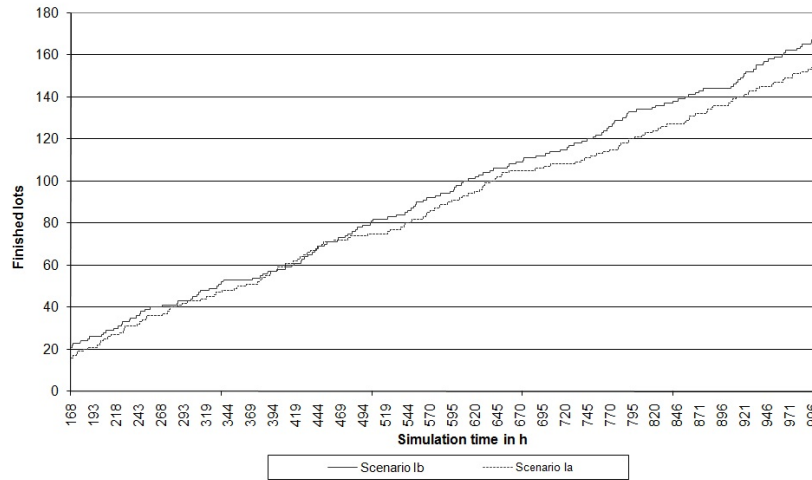


Fig. 20. Simulation results from change in prioritisation with scenario I. In scenario Ia three products in succession were prioritised, whereas in scenario Ib one distinct product was prioritised throughout the simulation.

Table 9 shows the WIP and finished lots at the end of each production week for scenario IIa and IIb. One can see clearly that the WIP is substantially higher in scenario IIb, so changing the prioritisation had a negative effect on overall productivity (the amount of finished work in scenario b is substantially lower and thus the work in progress higher). Figure 21 shows the development of production output graphically. It is remarkable that the production output in scenario IIb is not as steadily progressing as in scenario IIa. There are longer periods when

Tab. 9. Work in process (WIP) and finished lots in scenario II with a change in prioritisation

	WIP <i>Scenario IIa</i>	WIP <i>Scenario IIb</i>	Finished lots <i>Scenario IIa</i>	Finished lots <i>Scenario IIb</i>
Week 2	25	26	28	31
Week 3	31	39	29	27
Week 4	31	46	31	28
Week 5	37	59	31	27
Week 6	38	69	32	27

there is no production output at all. This can be explained by the utilisation of the processes: the resource demands for this scenario are quite extreme³¹. Since product 2 is prioritised, it occupies the stations in the beginning of the assembly line, while all other products have to wait. Product 4 is thus processed later than product 2. Now product 2 is running faster through the line, but product 4 has a longer production time. So after a while several products 2 are finished, but very few of product 4. They are still in the line, waiting in front of some resources. In the simulation with the basis prioritisation set this happens to a smaller extent, so the overall utilisation is higher in scenario IIa.

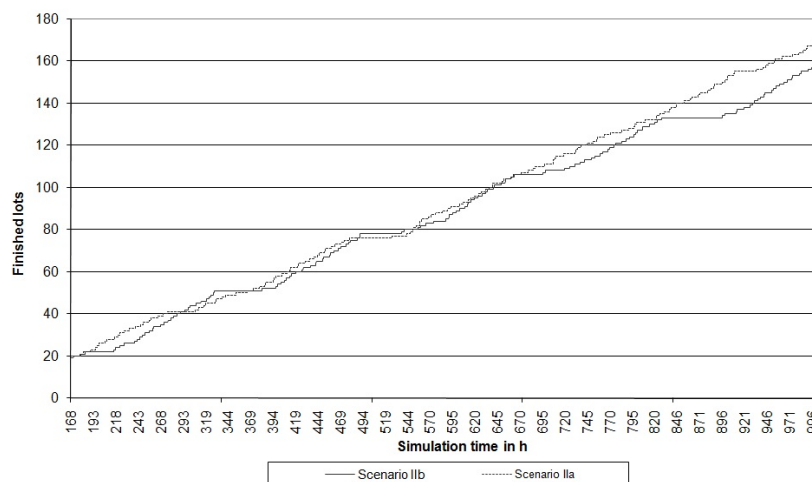


Fig. 21. Simulation results from change in prioritisation with scenario II

It should be noted that these were only static prioritisation policies. The same set of prioritisation was used throughout the simulation, regardless of its effectivity. Nevertheless, it becomes clear even with those two examples that such a static view can lead to undesired results.

Apparently, it is more promising to focus on idle resources: the two runs IIa and IIb give some hint for it. In scenario IIb the work allocation is worse than in

³¹Product four is the most time- and resource-consuming product, and its share in the product mix is comparatively high.

scenario IIa. The coordination of processes is much better in IIa. So a feasible and well-working alternative would be introducing feedback policies from the processes to the handling order. If there are products waiting in front of a working station those products that would prevent subsequent processes to become idle should be processed first. That could be achieved by various means: either implementing decision rules that are applied locally (for example a person that sets up detailed plans for each station or introducing a pull-system as in Kanban production) or monitoring the assembly line as a whole. Either way a flexible solution is much better than some deterministic rules, because it can be accounted for process deviances either caused by manual processing or by processing times of products that are not included into the product array in this model. A good prioritisation policy taking machine downtimes into consideration would be similar; in that case it has to be ensured that the production is not totally lean (some buffer lots should be in the line in order to prevent the machine to get idle).

5.2 Effect of scheduling

Scheduling is a policy where Osram has done some experimentations. A predominant mental model regarding scheduling was that it would substantially improve the output of the assembly line. There has been some research in the Operations Research community on this topic, but mainly with simple mathematical methods, showing positive effects of different scheduling policies. However, those assumptions about the benefits of scheduling for this system are challenged here.

A similar picture as in the case of prioritisation is obtained with the scheduling policy. Also in this case one positive and one negative example is presented. Table 10 shows the amount of products in scenario I and II.

The scheduling policy is introduced in order to relax the bottleneck at process one. So instead of introducing all products at once into the line they are split up: if, for example, 35 lots are introduced into the line scheduling means that a new product is put on the line every 4 hours and 50 minutes.³² The scheduling order is

³²In the model this can be easily done by using multiple pulse-functions.

Tab. 10. Selected scenarios for scheduling analysis

Scenario	I	II
Product 1	6	0
Product 2	4	13
Product 3	9	6
Product 4	8	4
Product 5	2	4
Total lots	35	38

kept constant throughout the simulation and different orders of products are tried out. The best scenario in this process is compared to the base run. In scenario Ia and IIa all lots are introduced at-once; scheduling is applied to scenario Ib and IIb.

Tab. 11. Work in process (WIP) and finished lots in scenario I with and without scheduling

	WIP <i>Scenario Ia</i>	WIP <i>Scenario Ib</i>	Finished lots <i>Scenario Ia</i>	Finished lots <i>Scenario Ib</i>
Week 2	26	26	31	29
Week 3	33	30	28	27
Week 4	27	31	30	31
Week 5	41	37	22	31
Week 6	45	31	30	33

Table 11 shows the positive effect of scheduling in scenario I. However, this effect is less in the case of changing the prioritisation, at least over the whole simulation time. Compare also figure 20 with figure 22. The dashed line for both graphs is the same. During the simulation, beginning with week 3 the changed prioritisation set is outperforming the base run clearly. Changing the scheduling policy is effecting the outcome not until week 5.

A negative effect of changing the scheduling policy is shown in scenario II (table 12). However, the difference is not very big in this case. Figure 23 pictures

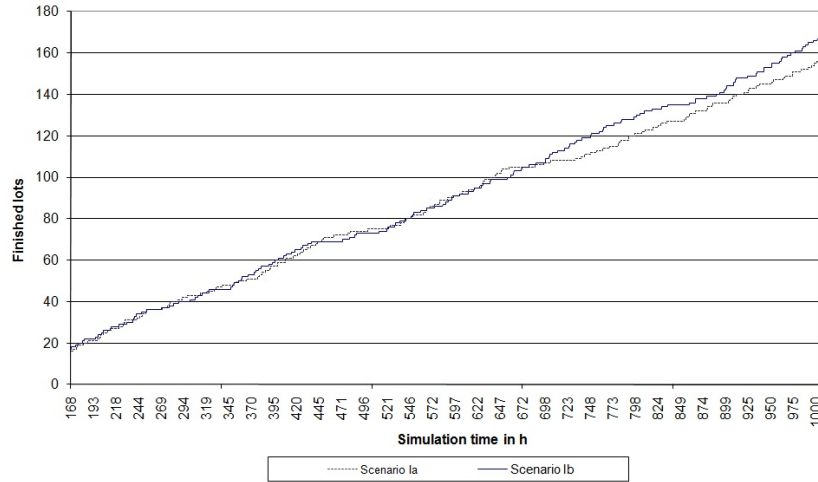


Fig. 22. Simulation results of scenario I with and without scheduling

the effect of scheduling for scenario II graphically.

Tab. 12. Work in process (WIP) and finished lots in scenario II with and without scheduling

	WIP <i>Scenario IIa</i>	WIP <i>Scenario IIb</i>	Finished lots <i>Scenario IIa</i>	Finished lots <i>Scenario IIb</i>
Week 2	9	10	29	27
Week 3	9	10	32	35
Week 4	11	9	29	26
Week 5	9	9	34	32
Week 6	12	11	28	28

Scenario IIa is the scenario with the highest overall output of all tested scenarios; it was found by static optimisation of the maximum possible output. 38 production lots is the calculated maximum output possible with the installed resources. The simulation clearly shows that this value is not obtained, there is still work in progress left. A static optimisation is thus not sufficient to calculate the output of the assembly line.

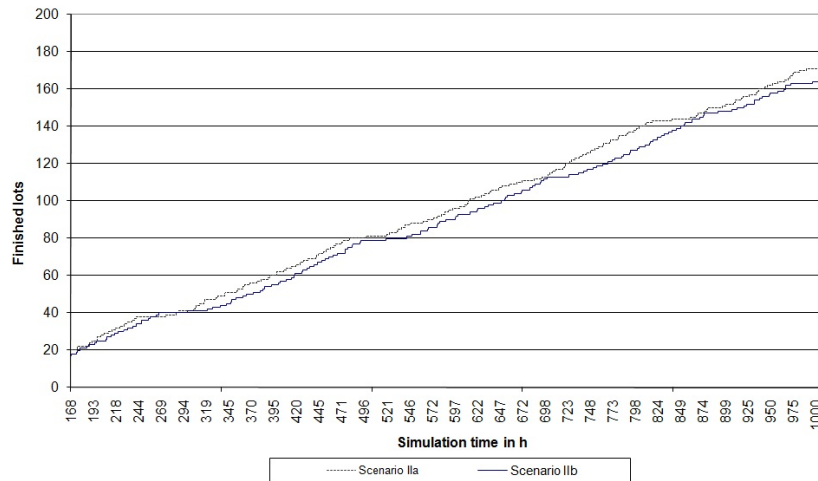


Fig. 23. Simulation results of scenario II with and without scheduling

The impact of a change in scheduling is smaller than the effect of introducing another prioritisation set. At a first glance this result is quite astonishing, regarding the amount of literature on scheduling topics compared to prioritisation. However, considering the tables shown in this and the previous section the reasons for it are quite clear: in all scenarios the output is somewhat smaller than the input. That means there is at least one process in the line where products have to wait. This is quite natural as the line is not balanced; problems related to line balancing are discussed in the next session. If the line is not balanced, there are products that are not processed immediately. The order of the processing of the working stations becomes more important than the order of products that are introduced into the line. So the longer the time horizon the more important prioritisation becomes in comparison to scheduling.

Changing the prioritisation, however, is only relevant for a distinct time span within the simulation. Over the whole time horizon of six weeks prioritisation policies are hardly outperforming scheduling policies. It is difficult to draw the line for the analysis: when should the simulation results be cut off? What is an appropriate time horizon? As the graphs already indicate: finding a good statistical parameter or analysis method is not obvious. Even dynamic deviance indicators

do not help in improving the analysis of the results. The cut-off of the simulation at some point in time seems to be arbitrary. And, in fact, evaluating the outcome solely on a graphical inspection basis may look unscientific, but in lack of better methods the only possibility.

The same problems as in the prioritisation case can be found in searching for the optimum solution for the order of products in the scheduling sequence. The number of options is too great for successfully applying optimisation methods. The exact number depends on the scenario³³; but it can be assumed that it is at least as big as in the prioritisation case.

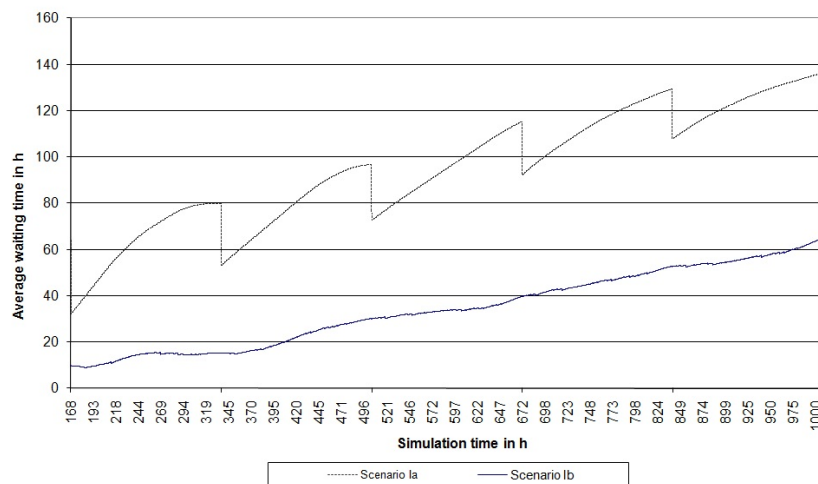


Fig. 24. Average waiting time of the products at process one (compare to the process flow in figure 2) in scenario I with and without scheduling. Process one has the highest overall waiting time in both scenarios. Since this is the process where product 1 to 4 is being processed first, scenario Ia has a substantially higher waiting time than scenario Ib. Note that in scenario Ia the average waiting time drops at the beginning of each week as new products enter the line.

Scheduling, though not as important as prioritisation for the line output can

³³The number of production lots and the partition in which they are divided is one factor influencing the total number of possible scenarios; furthermore in this case it may be feasible to change the scheduling in every week in order to increase the output.

have an effect on the number of lots waiting in line and thus on an important performance indicator that is used frequently. This is shown exemplarily for scenario I. Figure 24 pictures the average waiting time of all products of the working station with highest overall waiting time. Already in week 2 scenario Ia (the scenario without a scheduling policy) the average waiting time in hours is more than double as high as in scenario Ib.

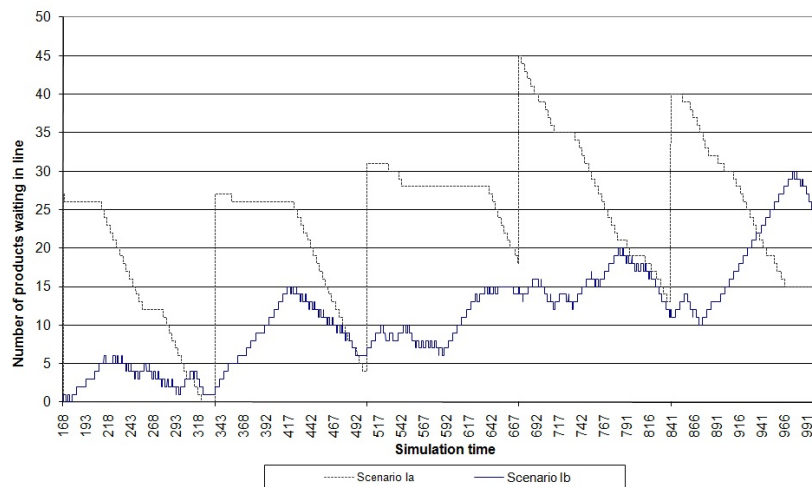


Fig. 25. Number of products waiting for being processed at process one in scenario I with and without scheduling. Process one is the process with the highest number of products waiting for both scenarios Ia and Ib. Note here that in scenario Ia at the beginning of each week new products are introduced, so the total number rises immediately.

Figure 25 shows the number of lots waiting in the line at process one. In this case it is the same process as in figure 24, but this fact is not a regularity. For some product mixes these two performance indicators differ. The number of products waiting is almost all the time higher for scenario Ia, compared to Ib. As a result, the cycle time is rising as well as the costs: more products waiting means more assets. The production with scheduling is leaner than the one without. So though the impact on overall output might not be too strong in the case of introducing a scheduling policy, it nevertheless has positive aspects that have to be considered in judging on its efficiency.

So far, only one possibility of setting up a scheduling policy was discussed. Scheduling, however, is one main tool in the Theory of Constraints (TOC) and in the following setting up a scheduling plan according to this theory is discussed.

The TOC (Gupta and Boyd, 2008) has its origins in the early 80s and was founded by Eliyahu Moshe Goldratt. It became quite popular, partly because its apparent simplicity, as some of the main works are written in a popular style using many analogies. However, this simplicity is rather superficial, a reason why there are not many successfully reported implementations in the literature (Mabin and Balderstone, 1998).

The idea behind it is that there is always one bottleneck within a production system and that all attention should be focused on that process. As process one is the bottleneck in scenario I an elaborate scheduling for it was implemented. This means that the products are scheduled according to their production time. For example, the introduction of the second product into the line is exactly at the point in time when the processing for product 1 is finished.

Results from incorporating the TOC are given in figure 26. Here the effects of different scheduling policies are pictured whereas the thick line represents an equal chopped scheduling and the other scheduling options (dashed and thinner lines) are made according to the Theory of Constraints.

One can observe that only in one case the scheduling according to the TOC is better than the base run with an equal chopped scheduling. The other scenarios perform either worse or as good as the base run. The difficulty with the TOC in this case is that process one is used also later in the production process. So a “perfect” scheduling would also depend upon taking subsequent processes into account. The scheduling planning is then as complex as in the ordinary scheduling and so the TOC is no help for easily improving the performance of the assembly line. However, differences are not big; they can hardly be distinguished.

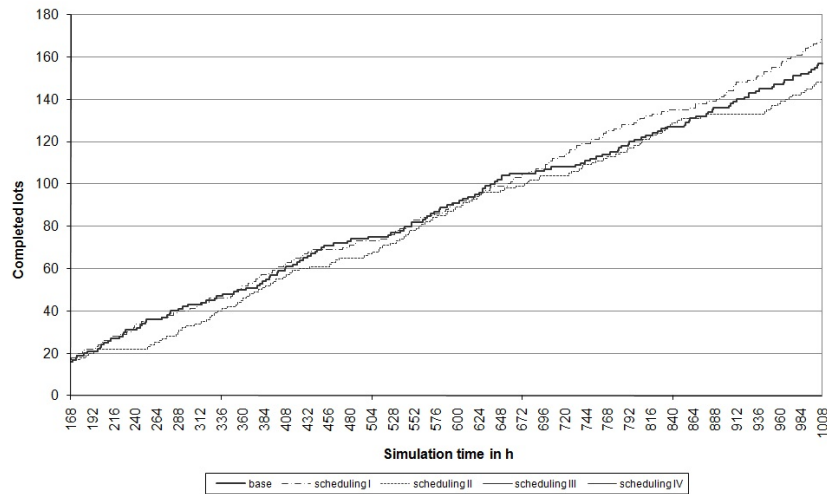


Fig. 26. Results from different scheduling policies according to the Theory of Constraints for scenario I. The basic feature of all scheduling options is the precise scheduling of products. The policies differ in the order of their products.

The result that scheduling with the TOC is not a major improvement confirms the theoretical findings from (Fung, 1999). He claimed that if the bottleneck is the first process, then a detailed scheduling would not result in a major output increase. In this case, some of the alternatives are even worse than an even chopped scheduling plan and only one is better. It is thus questionable whether or not the effort for setting up such a detailed scheduling plan as required by the TOC is economically reasonable.

5.3 Effect of bottleneck removal and line balancing

Detecting a bottleneck is not trivial (Leporis and Králová, 2010); there is no unique standard measure that could be applied. One could, e.g., define the bottleneck as the machine with the longest average waiting time for the products (Roser et al., 2001), but also the utilisation ratio or the number of lots waiting before one process could be taken. Most literature is focused on the best utilisation of machines; operators, who are also an integral factor in the production process are often neglected in the analysis of production systems (Baines et al., 2004).

The bottleneck of production depend upon *a*) the production mix, *b*) the question whether or not the line is balanced and *c*) the current state of the line. Bottleneck detection is closely related to line balancing. The goal of detecting the bottleneck of production is to remove it and to even out the different machines involved in the production process.

A balanced line is a fairly even usage of capacity within a production line. It is claimed to be a reasonable state as too much idle capacity is avoided, which was also one of the mental models prominent at Osram before this simulation project. There is a vast body of literature in the field of line balancing, using mainly mathematical models. But the simplification needed to find the optimal solution often hinders its realisation (Falkenauer, 2005).

Finding a bottleneck in a very imbalanced line is quite easy: if only one process is up all the time and all other processes have longer idle periods then the bottleneck of production is obvious. If, on the other hand, the line is perfectly balanced, then there is no single bottleneck at all; all processes are equally important and there is no laggard in the manufacturing sequence.

In the following one fairly imbalanced line was used in order to demonstrate the difficulties related to defining the bottleneck. This procedure is a repeated action: after having detected the bottleneck it will be removed. Then again the bottleneck is searched for and it is removed. However, this example shows that the closer one comes to a balanced line the more difficult it is *a*) to detect it and *b*) to remove it.

Scenario I of the scheduling analysis is applied in this example; table 13 lists again the product mix for this scenario; as the scheduling policy leads to an improvement of the overall output this policy is also applied in this analysis.

Figure 27 shows the result of the bottleneck removal procedure. Debottlenecking should be performed beginning with the last process in line (Sterman, 2000); here the procedure was successive: policy 1 brought a substantial improvement of

Tab. 13. Selected scenario for bottleneck analysis

Scenario	I
Product 1	6
Product 2	4
Product 3	9
Product 4	8
Product 5	2
Total lots	35

the output, so this setting was kept. Policy 2, 3 and 4 are extensions of policy 1, adding alternative resources to it.

The bottleneck in the base run was clearly at process one, so one resource was added to the line. However, the line was then quite balanced and the detection of the bottleneck became unclear. Neither the average waiting time before one process gave a definite result (policy 2), nor the average idle time of idle operators (policy 3), nor the products waiting before one process (policy 4). So adding a resource at process 1 (policy 2), one operator at cluster 3 (policy 3) or one resource at process 6 (policy 4) did not improve the performance of the line compared to policy 1. In the graph in figure 27 they are hardly distinguishable.

The output of the line could also not be increased by introducing more products (keeping the product mix ratio constant), so an overall system capacity constraint was reached after having introduced policy 1. The effectiveness of the selected policies (2-4) can also not be judged in terms of margin extra resource profitability, as their impact is indistinguishable. Obtaining a better result can only be obtained by adding multiple resources, as the balanced line does not indicate any clear bottleneck that can be removed.

Hence a balanced line has also some disadvantages. Those disadvantages have not been reported in the literature, although, in the light of this simulation study,

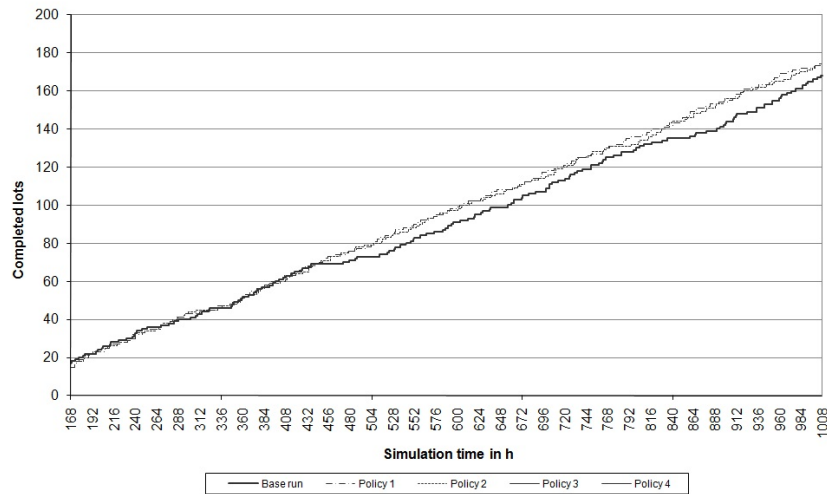


Fig. 27. *Different policies for bottleneck removal.* The base run is scenario I with a scheduling policy; policy option 2-4 are built upon policy 1: as policy one was effective, three further policy alternatives were tested.

they seem to be quite obvious.

Those disadvantages are:

- (a) defining a bottleneck in a balanced line is much harder than in an unbalanced one;
- (b) the balanced line only exists in a distinct production set within a multi-product assembly line, another product mix may cause the line being imbalanced again;
- (c) removing one bottleneck does not yield major output improvements;
- (d) judging the cost-effectiveness of a bottleneck removal in a balanced line is due to the little improvements almost impossible;
- (e) a balanced line can only be risen to a higher output level by investing at several stations within this line.

These effects have to be accounted for in order to judge the usefulness of line balancing in a particular case.

5.4 Practical implications of the results

The main findings gained from the simulation results are stated in the introduction of this chapter. These findings have several things in common:

- (I) **The insights gained are mostly qualitatively.** Neither a formula can be found that quantify some of the effects, nor would it make sense to come up with one, because of the next point.
- (II) **The different need of resources from the different products are paramount.** This does not only refer to the initial conditions, but also to all subsequent reflections on prioritisation, scheduling and bottleneck removal. The interactions resulting from the product mix are causing very different outcomes; not only from one scenario to the other, but also within the scenarios from one simulated week to the other.
- (III) **The interactions of the operators interlocking are essential for the product flow.** This is based upon the assumption that all operators are equally qualified within their cluster. In reality, the situation is much more complex.³⁴
- (IV) **The assembly line should be regarded as a whole.** In simulating the effects of the bottleneck removal as described in the previous section it became clear that in this case it is more promising to not concentrate on a single spot within the production line while judging its performance, but rather look at the system as a whole: this systemic view is supported by the ineffectiveness of removing the bottleneck in some situations. Partly, this view is already installed³⁵ by determining a product mix for every week locally for some crucial processes (though this determination is made on a static basis). It has been shown in this study that overloading the system leads to oscillatoric behaviour. A “pure” bottleneck analysis would not predict this behaviour.

³⁴This was partly examined during a one-week workshop independent from this study.

³⁵Though it may not be claimed as a systemic view, but the philosophy behind current management is implicitly systemic. However, by making this implicit assumption explicit some of the counter-intuitive effects of the system can be understood much better.

What are the implications of those findings on an effective management of the production line? First of all, the alignment of “soft” rules has to be accepted. Then a “systemic view” has to be implemented in the organisation. The management of information about the state of the system is crucial. However, within this complex system this is quite difficult. Therefore, the complexity for the actors in the system has to be reduced. The performance measures should be consistent. This is quite hard to achieve, since several stakeholders have different views on this system and thus have different opinions what is most important here. An example may serve as an illustration here.

What is the point of view upon the system?

Engineering point of view This point of view is based upon the utilisation of resources, which also touches the “ideal” of a balanced line. The main question within this point of view is: how do I produce the most with the resources I have?

Marketing point of view This point of view is focused on the realisation of a given product mix. The question here to be answered is: with which resources can I realise a defined product mix? A perfect realisation would often mean a re-engineering of the existing assembly line.

The management task here is to find a suitable agreement between both points of view. This simulation may help to achieve it, but other factors should not be ignored, such as the time needed for building up the capacity (which is very different for the different resources). That is in fact what became evident during the modelling exercise: a coherent specification of goals is missing. The point of view has to be clear and some rules of conduct are still mutually contradictory.

In sum, this study may help to adopt more of a systemic approach towards the management of this production line. There are nevertheless many points that could not have been touched in this study. For some of those aspects a softer problem solving approach might be better suited. It is this mixture of needing a detailed inclusion into the judgement of the system in combination with not ignoring non-measurable effects that makes the management of this system a challenging task.

This study is a first step towards a better understanding of the mechanisms that are prominent here.

6 Conclusions

This thesis examines one part of the assembly line of the laser production of Osram Semiconductors GmbH. System Dynamics (SD) was used as a problem structuring and simulation method; a simulation model of the assembly line was built. The goal of this study was to analyse three factors that are supposed to have an influence on overall production output. Those factors are *a)* the prioritisation of products, *b)* the scheduling of the different products in introducing them into the line and *c)* the detection and removal of the bottleneck of production in order to balance the line. The mental models widely held at Osram formed the hypotheses for this simulation project. Those mental models were:

- (a) Prioritisation is negative for the production output. A production that is based upon the first-in-first-out principle is more efficient than a production that violates this principle.
- (b) Scheduling is positive for the production output. Putting all products at once into the line should be avoided.
- (c) Detecting and removing the bottleneck of production is positive for the production output. Adding machines should substantially increase the line output.

In order to cope with the detailed complexity of the production with its various interactions discrete events had to be introduced in the simulation model. The discrete events could be introduced by classifying SD into the DTSS modelling formalism instead of the “classical” DESS modelling formalism (Zeigler, 1976). The timestep Δt used in every SD simulation software to perform the numerical analysis of the equations plays a crucial role and using it in the system’s equations allows to insert discrete entities. An array-structure was used to model the different products.

By using this method it enables to combine SD and Discrete Element Simulation (DES) without having to use hybrid approaches (Rabelo et al., 2005; Venkateswaran and Son, 2005). It is not necessary to build two separate simulation models and the problem of defining an interface or controller for ensuring the communication between the two platforms is avoided. However, the nature of chopping the time into even segments and not managing the simulation time with an event calendar like in DES software gives rise to some imprecision. The imprecision depends upon the choice of Δt and it is thus a trade-off between the accuracy of and the computing time for the simulation.

The model of the laser assembly line was developed in cooperation with various stakeholders of the system and it was validated in several workshops. The model used in this simulation study could have been built with a DES software, but taking SD software opens up for augmenting the time horizon of the simulation and including feedback effects that occur by taking other elements of the system into account. This is an advantage over DES where this is hardly possible.

The simulation results challenged the mental models at Osram: prioritisation, scheduling and bottleneck detection and removal were found to be strongly dependant on the state of the system and the composition of the product mix. The state of the system is defined by the products in the line, waiting for being processed. The product mix has a strong influence on the resources needed to perform the manufacturing processes. As both are constantly changing and also not totally predictable in the future, the system has to be flexible enough to react to a changing environment. Strict rules or formulas are insufficient as too many options would have to be considered. Instead, flexible rules depending on local feedback processes are much more promising.

It was demonstrated that the multiple use of resources with re-entrant flows in the production process is one of the main driver for the difficulties in predicting the production output. Static optimisation cannot be applied. It was shown that optimising prioritisation and scheduling would require high computation power and the profitability of that effort can be questioned in the light of the low

automatisation rate compared to the integrated circuits producing semiconductor industry.

Prioritisation had a stronger effect than scheduling due to the difficulty of setting up a “lean” assembly line with avoiding idle resources. For both cases, prioritisation and scheduling, supporting and contradicting examples for the mental models were found. Bottleneck removal is dependant on the balance of the line, which, in turn, is dependant on the production mix. So an even balanced line hinders a clear bottleneck detection and removing it does not substantially increase the production output. This has implications on the controlling system: the profitability of a new machine or of hiring new operators cannot be judged in absolute terms. It has always to take the desired product mix into consideration.

The time horizon of the simulation was restricted to six weeks in this study. Feedback effects of capacity constraints on backlog and sales and eventually on future demand as described by (Forrester, 1968) can thus be neglected. For a more holistic view of the laser production the time horizon should be augmented and the interactions with other departments should be included into the model. A more comprehensive understanding of the ongoing dynamics in the system and the installation of robust policies could be the gain.

References

- [Alfeld and Graham, 1976] Alfeld, L. and Graham, A. (1976). *Introduction to Urban Dynamics*. Wright-Allen Press.
- [Andersen, 2010] Andersen, D. (2010). Thinking about improvisation in Group Model Building sessions. Personal note on the presentation. Opening of Erasmus Mundus Master Programme in System Dynamics, Bergen, Norway.
- [Axelrod, 2006] Axelrod, R. (2006). Advancing the art of simulation in the social sciences. In Rennard, J.-P., editor, *Handbook of Research on Nature Inspired Computing for Economy and Management*. Hershey.
- [Axtell, 2010] Axtell, R. (2010). Pathologies of System Dynamics models or "why I am not a System Dynamicist". http://winforms.chapter.informs.org/presentation/Pathologies_of_System_Dynamics_Models-Axtell-20101021.pdf. Lecture at the Winforms.
- [Badal, 2006] Badal, M. (2006). Strategic management - problem structuring methods. Master's thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Lyngby, Denmark.
- [Bagdasaryan, 2010] Bagdasaryan, A. (2010). Systems theoretic techniques for modeling, control and decision support in complex dynamic systems. <http://arxiv.org/abs/1008.0775v1>.
- [Baines and Harrison, 1999] Baines, T. and Harrison, D. (1999). An opportunity for System Dynamics in manufacturing systems modeling. *Production Planning and Control*, 10(6):542–552.
- [Baines et al., 2004] Baines, T., Mason, S., Siebers, P.-O., and Ladbrook, L. (2004). Humans: the missing link in manufacturing simulation? *Simulation Modelling Practice and Theory*, 12(7-8):515–526.
- [Banks et al., 2005] Banks, J., Carson, J., Nelson, B., and Nicol, D. (2005). *Discrete - Event System Simulation*. Pearson, 4 edition.

-
- [Bartholdi et al., 2009] Bartholdi, J. J., Eisenstein, D. D., and Lim, Y. F. (2009). Deterministic chaos in a model of discrete manufacturing. *Naval Research Logistics (NRL)*, 56:293–299.
- [Bertrand and Fransoo, 2002] Bertrand, J. and Fransoo, J. (2002). Operations management research methodologies using quantitative modeling. *International Journal of Operations and Production Management*, 22(2):241–264.
- [Bezemer, 2003] Bezemer, J. (2003). Modelling the dynamics of business fulfilment: improved understanding of stacked lead time. Master’s thesis, TU Eindhoven, Technology Management.
- [Boysen et al., 2006] Boysen, N., Fliedner, M., and Scholl, A. (2006). A classification of assembly line balancing problems. Arbeits- und Diskussionspapiere der Wirtschaftswissenschaftlichen Fakultät der Friedrich-Schiller-Universität Jena, ISSN 1611-1311.
- [Brandolini et al., 2008] Brandolini, M., Briano, C., Briano, E., and Revetria, R. (2008). VV&A of complex modeling and simulation systems: Methodologies and case studies. *International Journal of Mathematics and Computers in Simulation*, 2(4):338–348.
- [Chahal and Eldabi, 2008] Chahal, K. and Eldabi, T. (2008). Which is more appropriate: a multi-perspective comparison between System Dynamics and Discrete Event Simulation. In *Proceedings of the European and Mediterranean Conference on Informations Systems, Dubai, United Arab Emirates*.
- [Chen and Jan, 2005] Chen, J. and Jan, T. (2005). A System Dynamics model of the semiconductor industry development in Taiwan. *The Journal of the Operational Research Society*, 56(10):1141–1150.
- [Chik et al., 2004] Chik, M., Ahmad, I., and Jamaludin, M. (2004). Paradigm shift in manufacturing operation through implementation of intelligent dispatching. In *Proceedings of the 39th Annual ORSNZ Conference, Auckland, NZ*.

-
- [Chwif et al., 2000] Chwif, L., Barretto, M., and Paul, R. (2000). On simulation model complexity. In *Proceedings of the 2000 Winter Simulation Conference, Orlando, FL*, pages 449–455.
- [Coyle, 1975] Coyle, R. (1975). Practical industrial applications of System Dynamics. *Dynamica*, 1:4–10.
- [Crist and Uzsoy, 2010] Crist, K. and Uzsoy, R. (2010). Prioritising production and engineering lots in wafer fabrication facilities: a simulation study. *International Journal of Production Research*, 49(11):3105–3125.
- [Dent, 2001] Dent, E. (2001). Systems science traditions: Differing philosophical assumptions. *Systems: Journal of Transdisciplinary Systems Science*, 6 (1-2):13–30.
- [Dijkstra, 1982] Dijkstra, E. (1982). On the role of scientific thought. In *Selected Writings on Computing: A Personal Perspective*. Springer-Verlag.
- [Dooley, 2002] Dooley, K. (2002). Organisational complexity. In Warner, M., editor, *International Encyclopedia of Business and Management*, pages 5013–5022. London: Thompson Learning.
- [Doyle and Ford, 1999] Doyle, J. and Ford, D. (1999). Mental models concepts revisited: some clarifications and a reply to Lane. *System Dynamics Review*, 15(4):411–415.
- [Dvergsdal, 2006] Dvergsdal, K. (2006). The emergency patient flow at Haukeland University hospital - a System Dynamics description and analysis. Master’s thesis, Department of Geography, System Dynamics Group, Faculty of Social Sciences, University of Bergen, Bergen, Norway.
- [Eden and Ackermann, 2006] Eden, C. and Ackermann, F. (2006). Where next for problem structuring methods. *The Journal of the Operational Research Society*, 57(7):766–768.
- [Epstein, 2008] Epstein, J. (2008). Why model? *Journal of Artificial Societies and Social Simulation*, 11(4):12–17.

-
- [Falkenauer, 2005] Falkenauer, E. (2005). Line balancing in the real world. In *Proceedings of the 2nd International Conference on Product Lifecycle Management PLM'05*, pages 360–370.
- [Färe and Primont, 1995] Färe, R. and Primont, D. (1995). *Multi-Output Production and Duality: Theory and Applications*. Kluwer Academic Publishers.
- [Filho and Uzsoy, 2008] Filho, M. and Uzsoy, R. (2008). Combining System Dynamics and Factory Physics approach to study the effect of continuous improvement on lot size - cycle time relationships. In *Proceedings of the 19th Annual Conference of the Production and Operations Management Society*.
- [Filho and Uzsoy, 2010] Filho, M. and Uzsoy, R. (2010). The effect of shop floor continuous improvement programs on the lot size-cycle time relationship in a multi-product single-machine environment. *The International Journal of Advanced Manufacturing Technology*, 52(5-8):669–681.
- [Fischhoff, 1976] Fischhoff, B. (1976). Attribution theory and judgment under uncertainty. In Harvey, J., I. W. K. R., editor, *New directions in attribution research*, volume 1. Hillsdale.
- [Forrester, 1961] Forrester, J. (1961). *Industrial Dynamics*. MIT Press.
- [Forrester, 1968] Forrester, J. (1968). Market growth as influenced by capital investment. *Industrial Management Review (MIT)*, 9:83–105.
- [Fowler and Rose, 2004] Fowler, J. and Rose, O. (2004). Grand challenges in modeling and simulation of complex manufacturing systems. *Simulation*, 80(9):469–476.
- [Frandsberg, 2003] Frandsberg, T. (2003). Systems Thinking - a study of alternatives of R. Flood, M. Jackson, W. Ulrich, and G. Midgley. *International Journal of Computers, Systems, and Signals*, 4(1):33–41.
- [Fritz, 1983] Fritz, R. (1983). A structural linguistics approach to method in System Dynamic modeling. In *Proceedings of the 1st International Conference of System Dynamics, Chestnut Hill, MA*.

-
- [Fung, 1999] Fung, K. (1999). Follow the laggard? - not all bottlenecks are created equal. *System Dynamics Review*, 15(4):403–410.
- [Godding et al., 2003] Godding, G., Sarjoughian, H., and Kempf, K. (2003). Semiconductor supply network simulation. In *Proceedings of the 2003 Winter Simulation Conference*, pages 1593–1601.
- [Goldwasser and Motwani, 1999] Goldwasser, M. H. and Motwani, R. (1999). Complexity measures for assembly sequences. *International Journal of Computational Geometry and Applications*, 9(4 & 5):371–417.
- [Gonçalves, 2003] Gonçalves, P. (2003). *Demand Bubbles and Phantom Orders in Supply Chains*. PhD thesis, Sloan School of Management, Massachusetts Institute of Technology.
- [Größler, 2007] Größler, A. (2007). Struktur und Dynamik strategischer Fähigkeiten in der Produktion. Habilitationsschrift, Mannheim University Press, Mannheim, Germany,.
- [Größler et al., 2008] Größler, A., Thun, J.-H., and Milling, P. (2008). System Dynamics as a structural theory in Operations Management. *Production and Operations Management*, 17(3):373–384.
- [Güneralp, 2004] Güneralp, B. (2004). A principle on structure-behavior relations in System Dynamics models. In *Proceedings of the 22nd International Conference of the System Dynamics Society, Oxford, England*.
- [Gupta and Boyd, 2008] Gupta, M. and Boyd, L. (2008). Theory of constraints: a theory for operations management. *International Journal of Operations and Production Management*, 28(10):991–1012.
- [Hao and Shen, 2008] Hao, Q. and Shen, W. (2008). Implementing a hybrid simulation model for a Kanban-based material handling system. *Robotics and Computer Integrated Manufacturing*, 24(5):635–646.
- [Haug and Schmitt-Rink, 1984] Haug, H. and Schmitt-Rink, S. (1984). Electron theory of the optical properties of laser-excited semiconductors. *Progress in Quantum Electronics*, 9:3–100.

-
- [Helal et al., 2007] Helal, M., Rabelo, L., Sepúlveda, J., and Jones, A. (2007). A methodology for integrating and synchronizing the System Dynamics and Discrete Event Simulation paradigms. In *Proceedings of the 25th International Conference of the System Dynamics Society, Boston, MA*.
- [Jacobson et al., 2008] Jacobson, J., Malczynski, L., and Tidwell, V. (2008). Very large System Dynamics models - lessons learned. In *Proceedings of the 26th International Conference of the System Dynamics Society, Athens, Greece*.
- [Jain et al., 1999] Jain, S., Lim, C., Gan, B., and Low., Y. (1999). Criticality of detailed modelling in semiconductor supply chain simulation. In *Proceedings of the 1999 Winter Simulation Conference, Phoenix, AZ*, pages 888–896.
- [Johnzén, 2009] Johnzén, C. (2009). *Modeling and optimizing flexible capacity allocation in semiconductor manufacturing*. PhD thesis, Ecole Nationale Supérieure des Mines de Saint-Etienne, Gardanne, France.
- [Kádár et al., 2004] Kádár, B., Pfeiffer, A., and Monostori, L. (2004). Discrete Event Simulation for supporting production planning and scheduling decisions in digital factories. In *Proceedings of the 37th CIRP International Seminar on Manufacturing Systems, Digital Enterprises, Production Networks, Budapest, Hungary*, pages 444–448.
- [Kampmann and Oliva, 2008] Kampmann, C. and Oliva, R. (2008). Structural dominance analysis and theory building in System Dynamics. *Systems Research and Behavioral Science*, 25:505–519.
- [Keating, 1998] Keating, E. (1998). Everything you ever wanted to know about how to develop a system dynamics model, but were afraid to ask. In *Proceedings of the 16th International Conference of the System Dynamics Society, Québec, Canada*.
- [Kogan, 2006] Kogan, K. (2006). Optimal scheduling of parallel machines with constrained resources. *European Journal of Operational Research*, 170:771–787.
- [Kreisler, 2005] Kreisler, B. (2005). Strategische Planung als Konstruktion von Wirklichkeit. Konstruktivismus versus Realismus in der strategischen Planung.

-
- Master's thesis, Alpen-Adria-Universität Klagenfurt, Fakultät für Wirtschaftswissenschaften und Informatik, Klagenfurt, Austria.
- [Lane, 2000] Lane, D. C. (2000). Should System Dynamics be described as a "hard" or "deterministic" systems approach? *Systems Research and Behavioral Science*, 17:3–22.
- [Law and Kelton, 1991] Law, A. and Kelton, D. (1991). *Simulation Modeling and Analysis*. McGraw-Hill.
- [Leporis and Králová, 2010] Leporis, M. and Králová, Z. (2010). A simulation approach to production line bottleneck analysis. In *Proceedings of the International Conference of Cybernetics and Informatics, Vyšňa Boca, Slovak Republic*.
- [Linge, 2007] Linge, C. (2007). Patient flow through the central surgery unit at Haukeland Universitetssykehus - a System Dynamic approach. Master's thesis, Department of Geography, System Dynamics Group, Faculty of Social Sciences, University of Bergen, Bergen, Norway.
- [Listl and Notzon, 2000] Listl, A. and Notzon, I. (2000). An operational application of System Dynamics in the automotive industry: Inventory management at BMW. In *Proceedings of the 18nd International Conference of the System Dynamics Society, Bergen, Norway*.
- [Mabin and Balderstone, 1998] Mabin, V. and Balderstone, S. (1998). A review of Goldratt's Theory of Constraints (TOC) - lessons from the international literature. In *Conference Proceedings of the 33rd annual conference of the Operational Research Society of New Zealand, Auckland, New Zealand*, pages 205–214.
- [Mahayuddin and Tjahjono, 2010] Mahayuddin, Z. and Tjahjono, B. (2010). Simulation modelling by classification of problems: a case of assembly lines. In *Proceedings of the OR Society Simulation Workshop, Worcestershire, England*.
- [Maine and Iliff, 1985] Maine, R. and Iliff, K. (1985). Identification of Dynamic Systems. Theory and Formulation. NASA Reference Publication 1138, NASA Center for AeroSpace Information, Hanover, MD.

- [Mason and Fowler, 2001] Mason, S. J. and Fowler, J. W. (2001). Maximizing delivery performance in semiconductor wafer fabs - potential implementation issues. In *Proceedings of the Industrial Engineering Research Conference, Dallas, TX*.
- [May and Spanos, 2006] May, G. and Spanos, C. (2006). *Fundamentals of Semiconductor Manufacturing and Process Control*. Wiley and Sons.
- [Minnich and Maier, 2007] Minnich, D. and Maier, F. (2007). Responsiveness and efficiency of pull-based and push-based planning systems in the high-tech electronics industry. In *Proceedings of the 25th International Conference of the System Dynamics Society, Boston, MA*.
- [Mönch et al., 2009] Mönch, L., Fowler, J., Dauzère-Pérès, S., Mason, S., and Rose, O. (2009). Scheduling semiconductor manufacturing operations: problems, solution techniques, and future challenges. In *Proceedings of the Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA 2009), Dublin, Ireland*, pages 192–201.
- [Morecroft, 1984] Morecroft, J. (1984). Strategy support models. *Strategic Management Journal*, 5(3):215–229.
- [Morecroft and Robinson, 2006] Morecroft, J. and Robinson, S. (2006). Comparing Discrete-Event Simulation and System Dynamics: Modelling a fishery. In *Proceedings of the 2006 OR Society Simulation Workshop*.
- [Mula et al., 2006] Mula, J., Poler, R., García-Sabater, J., and Lario, F. (2006). Models for production planning under uncertainty: A review. *International Journal of Production Economics*, 103:271–285.
- [O’Callaghan, 1986] O’Callaghan, R. (1986). A System Dynamic perspective on JIT-kanban. In *Proceedings of the 4th International Conference of the System Dynamics Society, Sevilla, Spain*, pages 959–1004.
- [Oliva, 2003] Oliva, R. (2003). Model calibration as a testing strategy for System Dynamics models. *European Journal of Operational Research*, 151:552–568.

-
- [Ossimitz and Mrotzek, 2008] Ossimitz, G. and Mrotzek, M. (2008). The basics of System Dynamics: Discrete vs. continuous modelling of time. In *Proceedings of the 26th International Conference of the System Dynamics Society, Athens, Greece*.
- [Owen et al., 2010] Owen, C., Albores, P., Greasley, A., and Love, D. (2010). Simulation in the supply chain context: matching the simulation tool to the problem. In *Proceedings of the OR Society Simulation Workshop, Worcestershire, England*.
- [Özgün and Barlas, 2009] Özgün, O. and Barlas, Y. (2009). Discrete vs. continuous simulation: When does it matter? In *Proceedings of the 27th International Conference of the System Dynamics Society, Albuquerque, NM*.
- [Potts and Kovalyov, 2000] Potts, C. and Kovalyov, M. (2000). Scheduling with batching: a review. *European Journal of Operational Research*, 120(2):228–249.
- [Rabelo et al., 2005] Rabelo, L., Helal, M., Jones, A., and Min, H.-S. (2005). Enterprise simulation: A hybrid system approach. *International Journal of Computer Integrated Manufacturing*, 18(6):498–508.
- [Rabelo et al., 2004] Rabelo, L., Helal, M., and Lertpattarapong, C. (2004). Analysis of Supply Chain using System Dynamics, Neural Nets, and Eigenvalues. In *Proceedings of the 2004 Winter Simulation Conference, Washington, DC*, pages 1136–1144.
- [Reisinger, 2009] Reisinger, C. (2009). Der Umsatzplanungsprozess bei der OS-RAM Opto Semiconductors GmbH. Strukturanalyse und Verbesserungspotentiale. Master’s thesis, Fachbereich Betriebswirtschaft der Hochschule Regensburg, Regensburg, Germany.
- [Robinson et al., 2003] Robinson, J., Fowler, J., and Neacy, E. (2003). Capacity loss factors in semiconductor manufacturing. Working Paper, Fabtime, San Jose, CA.

- [Rose, 2000] Rose, O. (2000). Why do simple wafer fab models fail in certain scenarios? In *Proceedings of the 2000 Winter Simulation Conference, Orlando, FL*, pages 1481–1490.
- [Roser et al., 2001] Roser, C., Nakano, M., and Tanaka, M. (2001). A practical bottleneck detection method. In *Proceedings of the 2001 Winter Simulation Conference, Arlington, VA*, pages 949–953.
- [Saleh and Davidsen, 2001] Saleh, M. and Davidsen, P. (2001). The origins of behavior patterns. In *Proceedings of the 19th International Conference of the System Dynamics Society, Atlanta, GA*.
- [Saleh and Myrtveit, 1999] Saleh, M. and Myrtveit, M. (1999). Designing a rational process for risk-taking. In *Proceedings of the 17th International Conference of the System Dynamics Society, Wellington, New Zealand*.
- [Schaffernicht, 2010] Schaffernicht, M. (2010). Causal loop diagrams between structure and behaviour: A critical analysis of the relationship between polarity, behaviour and events. *Systems Research and Behavioral Science*, 27:653–666.
- [Schömig, 2000] Schömig, A. (2000). OR Probleme in der Mikrochipfertigung. In *Proceedings of the Symposium on Operations Reserach, Dresden, Germany*.
- [Schwaninger and Hamann, 2005] Schwaninger, M. and Hamann, T. (2005). Theory-building with System Dynamics - principles and practices. Discussion paper No. 50, Institut für Betriebswirtschaftslehre, Universität St. Gallen, St. Gallen, Switzerland.
- [Semini et al., 2006] Semini, M., Fauske, H., and Strandhagen, J. (2006). Applications of Discrete-Event Simulation to support manufacturing logistics decision-making: a survey. In *Proceedings of the 2006 Winter Simulation Conference, Monterey, CA*.
- [Senge, 1990] Senge, P. M. (1990). *The Fifth Discipline. The Art and Practice of The Learning Organization*. Currency Doubleday, New York.
- [Simon, 1962] Simon, H. A. (1962). The architecture of complexity. *Proceedings of the American Philosophical Society*, 106(6):467–482.

-
- [Speller et al., 2007] Speller, T., Rabelo, L., and Jones, A. (2007). Value chain modelling using System Dynamics. *International Journal of Manufacturing Technology and Management*, 11(2):135–156.
- [Sterman, 1991] Sterman, J. (1991). A skeptic’s guide to computer models. In et al., G. B., editor, *Managing a Nation: The Microcomputer Software Catalog*, pages 209–229. Westview Press, Boulder, CO.
- [Sterman, 2000] Sterman, J. (2000). *Business Dynamics: Systems Thinking and Modeling for a Complex World*. McGraw-Hill.
- [Sterman, 2002] Sterman, J. (2002). All models are wrong: reflections on becoming a systems scientist. *System Dynamics Review*, 18(4):501–531.
- [Sweetser, 1999] Sweetser, A. (1999). A comparison of System Dynamics (SD) and Discrete Event Simulation (DES). In *Proceedings of the 17th International Conference of the System Dynamics Society, Wellington, New Zealand*.
- [Tako and Robinson, 2008] Tako, A. and Robinson, S. (2008). Model-building in System Dynamics and Discrete Event Simulation: a quantitative comparison. In *Proceedings of the 26th International Conference of the System Dynamics Society, Athens, Greece*.
- [Thillainathan, 1975] Thillainathan, V. (1975). Capacity acquisition for continuous systems. *Dynamica*, 1:11–18.
- [Vaneman and Triantis, 2003] Vaneman, W. and Triantis, K. (2003). The dynamic production axioms and System Dynamics behaviors: The foundation for future integration. *Journal of Productivity Analysis*, 19:93–113.
- [Vangheluwe et al., 2002] Vangheluwe, H., de Lara, J., and Mosterman, P. (2002). An introduction to multi-paradigm modelling and simulation. In *Proceedings of the AIS’2002 Conference (AI, Simulation and Planning in High Autonomy Systems), Lisboa, Portugal*.
- [Vázquez and Liz, 2007] Vázquez, M. and Liz, M. (2007). System Dynamics and philosophy. a constructivist and expressivist approach. In *Proceedings of the 25th International Conference of the System Dynamics Society, Boston, MA*.

-
- [Venkateswaran and Son, 2005] Venkateswaran, J. and Son, Y.-J. (2005). Hybrid system dynamic - discrete event simulation-based architecture for hierarchical production planning. *International Journal of Production Research*, 43(20):4397–4429.
- [Vennix et al., 1996] Vennix, J., Akkermans, H., and Rouwette, E. (1996). Group model-building to facilitate organizational change: an exploratory study. *System Dynamics Review*, 12(1):39–58.
- [Visser, 2007] Visser, M. (2007). System Dynamics and group facilitation: contributions from communication theory. *System Dynamics Review*, 23(4):453–463.
- [Wagner, 2004] Wagner, R. (2004). *Stock-Flow-Thinking und Bathtub Dynamics. Eine Theorie von Bestands- und Flussgrößen*. PhD thesis, Alpen-Adria-Universität Klagenfurt, Fakultät für Wirtschaftswissenschaften und Informatik, Klagenfurt, Austria.
- [Wolstenholme, 1983] Wolstenholme, E. (1983). System Dynamics: A system methodology or a system modelling technique. *Dynamica*, 9:84–90.
- [Wu, 2005] Wu, D. (2005). *Unified frameworks for optimal production planning and scheduling: continuous-time decomposition-based approaches*. PhD thesis, Graduate School-New Brunswick, Rutgers, the State University of New Jersey, New Brunswick, NJ.
- [Wu, 2007] Wu, K. (2007). Modeling the semiconductor industry dynamics. Master’s thesis, Computation for Design and Optimization, School of Engineering, Massachusetts Institute of Technology.
- [Zeigler, 1976] Zeigler, B. (1976). *Theory of Modelling and Simulation*. Wiley.
- [Zeigler et al., 2000] Zeigler, B., Prähofer, H., and Gon Kim, T. (2000). *Theory of Modeling and Simulation: Integrating Discrete Events and Continuous Complex Dynamic Systems*. Academic Press, 2nd edition.

A Model documentation

Five products are incorporated into the model. Each of them takes a different route through the system and utilises different resources, both the processing times and the times for the operators differ between the products. The process flow of products is shown graphically in figure 2 on page 6.

- Product one: p (process) 1, p3, p6-1, p7, p8, p14, p10, p13, p16
- Product two: p1, p3/p4³⁶, p6-1, p7, p8/p9, p14, p10, p13, p16
- Product three: p1, p3, p6-1, p7, p8, p10, p12, p13, p14, p13, p16
- Product four: p1, p3, p5, p6-1/p6-3³⁷, p7, p8, p10, p12, p13, p14, p13, p16
- Product five: p2, p3, p6-1/p6-3, p7, p8, p11, p15, p13, p16

Process 3, 7, 8, 13 and 16 are purely manual.³⁸ Process 1 and 10 cannot be run simultaneously on the same machines, the same applies to process 1 and 2; p2 and 11; p3 and 4; p13 is sometimes run twice (in figure 2 this is divided into p13 and p13b) in the flow, so there is also a process conflict. Furthermore, there is a lot split for product one and two after p1. Product four is run at a batch process for p5 and 6, i.e., two lots are processed at the same time and before p16 two lots of product four are summarised into one.

There are four clusters of operators; each cluster of operators has its own specialisation. Cluster one is able to process p1-4 and p7-11; cluster two p5 and 6; cluster three p12-15 and cluster four only p16.

³⁶Product two can either run on process 3 or 4 which has to be decided before starting the simulation. The same applies to p8 and p9.

³⁷p6-1 and p6-2 are duplicates, product four and five can run either on process p6-1 or on process p6-3. This is decided according to the products waiting in front of p6-1 and p6-3, respectively.

³⁸Process three requires in addition to that three operators simultaneously.

A MODEL DOCUMENTATION

The model is initialised with no products in line; the unit of measurement is production lots and hours, the simulation time is set to 1008 hours, which is equivalent to 6 weeks.

B Model equations

The whole model is too huge to list here the equations of all of the variables.³⁹ Instead, the equations of the model shown in figure 10 on page 46 are listed here; this basic building block is applied to the processes as shown in figure 2 on page 6 and listed in appendix A.

This documentation is in alphabetic order and built up as followed:

- (I) name of the variable;
- (II) definition of the variable (equation for auxillary and flows; equation, initial value and in- and outflows for the stocks);
- (III) type of variable (aux: auxillary, const: constant, stock, logical);
- (IV) unit of the variable (operators, product unit, hour,..., unitless);
- (V) dimension of the variable (Fifo queue: $\{1 \dots 10\}$ ⁴⁰, Fifo shifting range: FIRST ('FIFO queue') . . . LAST('FIFO queue')-1}, operating time: {'machine time', 'operator time'}, Products: {Prod1,Prod2,Prod3,Prod4,Prod5}, $1 \dots 3$ ⁴¹). No indication means the dimension is one.

The model was built with Powersim Studio 8 Academic, Service Release 5a, from Powersim Software AS.

Allocate idle operators PRIORITYALLOCDISCRETE(INTEGER('operators idle'), {'request one','request two','request three'}, {'priority process one and two','priority process one and two','priority process two'}, FALSE) *auxillary, operators, 1 . . . 3*

³⁹The final model contains 1.089 variables, 15 global units and 6 global ranges.

⁴⁰In the full model the numerical subrange has to be defined in dependence of the maximum numbers of products waiting before one process; if this dimension is underdefined the model will not produce meaningful results. If, on the other hand, the range is too large the calculation time suffers significantly in large models.

⁴¹This dimension is variable and depends on the number of resources in one operator cluster on which the operators should be allocated.

aux op idle p1 IF('aux operator idle p1'<>0 AND 'aux op idle p1 process stop'=0 , 1«'product unit'»/TIMESTEP) *aux, product unit/hour*

aux op idle p1 process stop DELAYPPL('aux operator idle p1',TIMESTEP)
aux, unitless

aux op idle p2 IF('aux operator idle p2'<>0 AND 'aux op idle p2 process stop'=0 , 1«'product unit'»/TIMESTEP) *aux, product unit/hour*

aux op idle p2 process stop DELAYPPL('aux operator idle p2',TIMESTEP)
aux, 1«'product unit'»/TIMESTEP aux, unitless

aux op idle p3 IF('aux operator idle p3'<>0 AND 'aux op idle p3 process stop'=0 , 1«'product unit'»/TIMESTEP) *aux, product unit/hour*

aux op idle p3 process stop DELAYPPL('aux operator idle p3',TIMESTEP)
aux, unitless

aux operator idle p1 IF(LOOKUP('process time process one',2)=0 , 1) *aux, unitless*

aux operator idle p2 IF(LOOKUP('process time process two',2)=0 , 1) *aux, unitless*

aux operator idle p3 IF(LOOKUP('process time process three',2)=0 , 1) *aux, unitless*

aux ps set idle process one DELAYPPL (LOOKUP('reverse process time process one',2),TIMESTEP) *aux, 1/hour*

aux ps set idle process three DELAYPPL (LOOKUP ('reverse process time process three', 2), TIMESTEP) *aux, 1/hour*

aux ps set idle process two DELAYPPL (LOOKUP('reverse process time process two',2),TIMESTEP) *aux, 1/hour*

Buffer stock 0«'product unit'», outflow { COLLECT('Start production process one and two') }, outflow {COLLECT('Start production process three') }, inflow {'set in' } *stock, product unit, Products*

decision open way IF(ARRSUM('Sum fifo p1-2')>ARRSUM('Sum fifo p3'),1, IF(ARRSUM('Sum fifo p1-2')=ARRSUM('Sum fifo p3'), IF(NUMBER(LOOKUP('Production time process one and two',INTEGER(SCANGT('open way auxillary',0))))>=NUMBER(LOOKUP('Production time process three',INTEGER(SCANGT('open way auxillary',0))))),1, IF(ARRSUM('Buffer stock'*'Production time process one and two')>=ARRSUM('Buffer stock'*'Production time process three'),1,0))) **documentation:**
1 for process one and two, 0 for process three *aux, unitless*

expired time process one IF('reverse process time process one'*TIMESTEP>'process time process one', 'process time process one' / TIMESTEP, 'reverse process time process one') *aux, 1/hour, 'operating time'*

expired time process three IF('reverse process time process three'*TIMESTEP>'process time process three', 'process time process three' / TIMESTEP, 'reverse process time process three') *aux, 1/hour, 'operating time'*

expired time process two IF('reverse process time process two'*TIMESTEP>'process time process two', 'process time process two' / TIMESTEP, 'reverse process time process two') *aux, 1/hour, 'operating time'*

fillup time process one IF(ARRSUM('process time process one')=0,'Process time deviation'/TIMESTEP,0«1/hour») *aux, 1/hour, 'operating time'*

fillup time process three IF(ARRSUM('process time process three')=0,'Process time deviation'/TIMESTEP,0«1/hour») *aux, 1/hour, 'operating time'*

fillup time process two IF(ARRSUM('process time process two')=0,'Process time deviation'/TIMESTEP,0«1/hour») *aux, 1/hour, 'operating time'*

Finished process one 0«'product units'», inflow {'Production rate process one'}
stock, product unit, Products

Finished process three 0«'product units'», inflow {'Production rate process three'}
stock, product unit, Products

Finished process two 0«'product units'», inflow {'Production rate process two'}
stock, product unit, Products

Index of most urgent product p1-2 IF('Priority processing p1-2', SCANEQ('Sum waiting among oldest p1-2', ARRMAX('Sum waiting among oldest p1-2')), SCANGT('Priority product',0)) *aux, unitless*

Index of most urgent product p3 IF('Priority processing p3', SCANEQ('Sum waiting among oldest p3', ARRMAX('Sum waiting among oldest p3')), SCANGT('Priority product',0)) *aux, unitless*

Index of oldest element in fifo queue p1-2 IF('Priority processing p1-2', SCANGT('Sum fifo p1-2', 0«'product units'», TRUE), SCANGT('Index of priority product in line p1-2', 0«'product units'», TRUE)) *aux, unitless*

Index of oldest element in fifo queue p3 IF('Priority processing p3', SCANGT('Sum fifo p3', 0«'product units'», TRUE), SCANGT('Index of priority product in line p3', 0«'product units'», TRUE)) *aux, unitless*

Index of priority product in line p1-2 IF(ARRSUM('Priority product')<>0 , LOOKUP('Waiting to start production process one and two', INTEGER (SCANGT('Priority product',0))) ARRSUM('Waiting to start production process one and two',1,1)) *aux, product unit, Fifo queue*

Index of priority product in line p3 IF(ARRSUM('Priority product')<>0 , LOOKUP('Waiting to start production process three', INTEGER(SCANGT ('Priority product',0))) , ARRSUM('Waiting to start production process three',1,1)) *aux, product unit, Fifo queue*

Needed operators for production process 1«operator/'product unit'» *const, operator/product unit*

Oldest products waiting p1-2 'Waiting to start production process one and two'[* ,INDEX('Index of oldest element in fifo queue p1-2')] *aux, product unit, Products*

Oldest products waiting p3 'Waiting to start production process three' [* ,INDEX('Index of oldest element in fifo queue p3')] *aux, product unit, Products*

open way auxillary IF('Policy switch process three down'=0 , IF(ABS('Production time process one and two'-'Production time process three')<>('Production time process one and two'+'Production time process three') , 1))
aux, unittless, Products

operator process one 0«operator», inflow {'operator process one inflow'}, outflow {'operator process one outflow'} *stock, operator*

operator process one inflow ARRSUM('Start new production process one')*
 'Needed operators for production process'/TIMESTEP *aux, operator/hour*

operator process one outflow IF('operator process one'>0«operator» AND ('Process stop set idle process one'<> 0«'product unit'/hour» OR 'aux op idle p1'<>0«'product unit'/hour») , ('Process stop set idle process one'+'aux op idle p1')* 'Needed operators for production process') *aux, operator/hour*

operator process three 0«operator», inflow {'operator process three inflow'}, outflow {'operator process three outflow'} *stock, operator*

operator process three inflow ARRSUM('Start new production process three')*
 'Needed operators for production process'/TIMESTEP *aux, operator/hour*

operator process three outflow IF('operator process three'>0«operator» AND ('Process stop set idle process three'<> 0«'product unit'/hour» OR 'aux op idle p3'<>0«'product unit'/hour») , ('Process stop set idle process three'+'aux op idle p3')* 'Needed operators for production process') *aux, operator/hour*

operator process two 0«operator». inflow {'operator process two inflow'}, outflow {'operator process two outflow'} *stock, operator*

operator process two inflow ARRSUM('Start new production process two')*
 'Needed operators for production process'/TIMESTEP *aux, operator/hour*

operator process two outflow IF('operator process two'>0«operator» AND ('Process stop set idle process two'<> 0«'product unit'/hour» OR 'aux op idle p2'<>0«'product unit'/hour») , ('Process stop set idle process two'+'aux op idle p2')* 'Needed operators for production process') *aux, operator/hour*

Operator time process one and two {1.8,1.8,1.8,1.8,0} «hour/'product unit'»
const, hour/product unit

Operator time process three {0,0,0,1.8,1.8} «hour/'product unit'» *const,*
hour/product unit

operators busy 0«operator», inflow {'set operators busy'}, outflow {'set operators idle'} *stock, operator*

operators idle 2«operator», outflow {'set operators busy'}, inflow {'set operators idle'} *stock, operator*

Policy switch process one down IF(NUMBER(TIME)>='Start for downtime p one' AND NUMBER(TIME)<='Stop for downtime p one' AND ('Start for downtime p one' + 'Stop for downtime p one')<>0 , 1) *aux, unitless*

Policy switch process three down IF(NUMBER(TIME)>='Start for downtime p three' AND NUMBER(TIME)<='Stop for downtime p three' AND ('Start for downtime p three'+ 'Stop for downtime p three')<>0 , 1) *aux, unitless*

Policy switch process two down IF(NUMBER(TIME)>='Start for downtime p two' AND NUMBER(TIME)<='Stop for downtime p two' AND ('Start for downtime p two' + 'Stop for downtime p two')<>0 , 1) *aux, unitless*

priority process one and two NUMBER(ARRSUM('Sum waiting each product p1-2')) *aux, unitless*

priority process two NUMBER(ARRSUM('Sum waiting each product p3')) *aux, unitless*

Priority processing p1-2 ARRSUM('Priority product'*'Sum waiting each product p1-2')=0«'product unit'» *logical*

Priority processing p3 ARRSUM('Priority product'*'Sum waiting each product p3')=0«'product unit'» *logical*

Priority product {0,1,0,0,0} *const, unitless, Products*

Process delay ARRSUM('Start new production process one')<>0«'product unit'»
logical

Process stop set idle process one IF('Policy switch process one down'=1
AND LOOKUP('reverse process time process one',2)<>'aux ps set idle pro-
cess one' AND LOOKUP('process time process one',2)>0 , 1«'product unit'»/
TIMESTEP) *aux, product unit/hour*

Process stop set idle process three IF('Policy switch process three down'=1
AND LOOKUP('reverse process time process three',2)<>'aux ps set idle
process three' AND LOOKUP('process time process three',2)>0 , 1«'product
unit'»/TIMESTEP) *aux, product unit/hour*

Process stop set idle process two IF('Policy switch process two down'=1 AND
LOOKUP('reverse process time process two',2)<>'aux ps set idle process
two' AND LOOKUP('process time process two',2)>0 , 1«'product unit'»/-
TIMESTEP) *aux, product unit/hour*

Process time deviation NORMAL(1,0.1) *aux, unitless*

process time process one 1, outflow {'expired time process one'}, inflow {'fillup
time process one'} *stock, unitless, 'operating time'*

process time process three 1, outflow {'expired time process three'}, inflow
{'fillup time process three'} *stock, unitless, 'operating time'*

process time process two 1, outflow {'expired time process two'}, inflow {'fillup
time process two'} *stock, unitless, 'operating time'*

Production in progress process one 0«'product units'», outflow {'Production
rate process one'}, inflow {COLLECT('Start new production process one')}
stock, product unit, Products

Production in progress process three 0«'product units'», inflow {COLLECT
('Start new production process three')}, outflow {'Production rate process
three'} *stock, product unit, Products*

Production in progress process two 0«product units», inflow {COLLECT ('Start new production process two')}, outflow {'Production rate process two'} *stock, product unit, Products*

Production rate process one IF(ARRSUM('process time process one')=0 AND ARRSUM('Production in progress process one')<>0«product unit» , 'Production in progress process one'/TIMESTEP , 0«product unit»/TIMESTEP 0«product unit»/TIMESTEP) *aux, product unit/hour, Products*

Production rate process three IF(ARRSUM('process time process three')=0 AND ARRSUM('Production in progress process three')<>0«product unit» , 'Production in progress process three'/TIMESTEP , 0«product unit»-/TIMESTEP) *aux, product unit/hour, Products*

Production rate process two IF(ARRSUM('process time process two')=0 AND ARRSUM('Production in progress process two')<>0«product unit» , 'Production in progress process two'/TIMESTEP , 0«product unit»/TIMESTEP) *aux, product unit/hour, Products*

Production time process one and two 2,2,2,1.8,0«hour/'product unit'» *const, hour/product unit, Products*

Production time process three 0,0,0,2,2«hour/'product unit'» *const, hour/product unit, Products*

production way one 'Production time process one and two'-('Production time process one and two'*'open way auxillary') *aux, hour/product unit, Products*

production way two 'Production time process three'-('Production time process three'*'open way auxillary') *aux, hour/product unit, Products*

Ready to start new production process one ARRSUM('Production rate process one'*TIMESTEP) >= ARRSUM('Production in progress process one') AND LOOKUP('Allocate idle operators',1) > 0«operator» AND 'Policy switch process one down' = 0 *logical*

Ready to start new production process three $\text{ARRSUM}(\text{'Production rate process three'} * \text{TIMESTEP}) \geq \text{ARRSUM}(\text{'Production in progress process three'})$ AND $\text{LOOKUP}(\text{'Allocate idle operators'}, 3) > 0$ «operator» AND 'Policy switch process three down' = 0 *logical*

Ready to start new production process two $\text{ARRSUM}(\text{'Production rate process two'} * \text{TIMESTEP}) \geq \text{ARRSUM}(\text{'Production in progress process two'})$ AND $\text{LOOKUP}(\text{'Allocate idle operators'}, 2) > 0$ «operator» AND 'Policy switch process two down' = 0 *logical*

request one IF($\text{ARRSUM}(\text{'Production in progress process one'}) = 0$ «'product unit'») AND $\text{ARRSUM}(\text{'Sum fifo p1-2'}) > 0$ «'product unit'» AND 'Policy switch process one down' = 0 , $\text{INTEGER}(\text{'Needed operators for production process'}) * 1$ «'product unit'») *aux, operator*

request three IF($\text{ARRSUM}(\text{'Production in progress process three'}) = 0$ «'product unit'») AND $\text{ARRSUM}(\text{'Sum fifo p3'}) > 0$ «'product unit'» AND 'Policy switch process three down' = 0 , $\text{INTEGER}(\text{'Needed operators for production process'}) * 1$ «'product unit'») *aux, operator*

request two IF($\text{ARRSUM}(\text{'Production in progress process two'}) = 0$ «'product unit'») AND $\text{ARRSUM}(\text{'Sum fifo p1-2'}) > 0$ «'product unit'» AND 'Policy - switch process two down' = 0 , $\text{INTEGER}(\text{'Needed operators for production process'}) * 1$ «'product unit'») *aux, operator*

reverse machine time process one $1 \text{ DIVZ0 } \text{ARRSUM}(\text{'Production time process one and two'} * \text{'Production in progress process one'})$ *aux, 1/hour*

reverse machine time process three $1 \text{ DIVZ0 } \text{ARRSUM}(\text{'Production time process three'} * \text{'Production in progress process three'})$ *aux, 1/hour*

reverse machine time process two $1 \text{ DIVZ0 } \text{ARRSUM}(\text{'Production time process one and two'} * \text{'Production in progress process two'})$ *aux, 1/hour*

reverse operator time process one $1 \text{ DIVZ0 } \text{ARRSUM}(\text{'Operator time process one and two'} * \text{'Production in progress process one'})$ *aux, 1/hour*

reverse operator time process three 1 DIVZ0 ARRSUM('Operator time process three'*'Production in progress process three') *aux, 1/hour*

reverse operator time process two 1 DIVZ0 ARRSUM('Operator time process one and two'*'Production in progress process two') *aux, 1/hour*

reverse process time process one FOR(p='operating time' | IF('Policy switch process one down'=1, 0 «1/hour», {'reverse machine time process one', 'reverse operator time process one'}[p])) **Documentation: 1/x, where x is the time needed for each product to process *aux, 1/hour, operating time***

reverse process time process three FOR(p='operating time' | IF('Policy switch process three down'=1, 0 «1/hour», {'reverse machine time process three', 'reverse operator time process three'}[p])) **Documentation: 1/x, where x is the time needed for each product to process *aux, 1/hour, operating time***

reverse process time process two FOR(p='operating time' | IF('Policy switch process two down'=1, 0 «1/hour», {'reverse machine time process two', 'reverse operator time process two'}[p])) **Documentation: 1/x, where x is the time needed for each product to process *aux, 1/hour, operating time***

set in PULSE({0,0,0,1,0}«'product unit'»,STARTTIME,9999«hour») *aux, product unit/hour, Products*

Set one product each process one and two IF(('open way auxillary'*'Buffer stock')>0«'product unit'» AND 'decision open way'=0 , IF('Buffer stock'>1«'product unit'», ('Buffer stock'-('Buffer stock'-1«'product unit'»))* ('Production time process one and two' DIVZ0 'Production time process one and two') , 'Buffer stock'* ('Production time process one and two' DIVZ0 'Production time process one and two')) , IF('Buffer stock'>1«'product unit'», ('Buffer stock'-('Buffer stock'-1«'product unit'»))* ('production way one' DIVZ0 'production way one') , 'Buffer stock'* ('production way one' DIVZ0 'production way one'))) *aux, product unit, Products*

Set one product each process three IF(('open way auxillary'*'Buffer stock') >0«'product unit'» AND 'decision open way'=1 , IF('Buffer stock' >1«'product unit'», ('Buffer stock'-('Buffer stock'-1«'product unit'»))* ('Production time process three' DIVZ0 'Production time process three') , 'Buffer stock'* ('Production time process three' DIVZ0 'Production time process three')) , IF('Buffer stock'>1«'product unit'», ('Buffer stock'-('Buffer stock'- 1«'product unit'»))* ('production way two' DIVZ0 'production way two') , 'Buffer stock'* ('production way two' DIVZ0 'production way two'))) *aux, product unit, Products*

set operators busy 'operator process one inflow' + 'operator process two inflow' + 'operator process three inflow' *aux, operator/hour*

set operators idle 'operator process one outflow' + 'operator process two outflow' + 'operator process three outflow' *aux, operator/hour*

Shift fifo process one and two FOR(p=Products,i='FIFO shifting range' | IF(SIGN(ARRSUM('Start production process one and two')) > 0, 'Waiting to start production process one and two'[p,i]-'Start new production process one'[p,i]-'Start new production process two'[p,i])) *aux, product unit, {Products, 'FIFO shifting range'}, zero order integration*

Shift fifo process three FOR(p=Products,i='FIFO shifting range' | IF(SIGN(ARRSUM('Start production process three')) > 0, 'Waiting to start production process three'[p,i]-'Start new production process three'[p,i])) *aux, product unit, {Products, 'FIFO shifting range'}, zero order integration*

Start for downtime p one 0 *const, unitless*

Start for downtime p three 0 *const, unitless*

Start for downtime p two 0 *const, unitless*

Start new production process one FOR(p=Products,i='FIFO queue' | IF(('Ready to start new production process one' AND NUMERICAL(p) = 'Index of most urgent product p1-2' AND NUMERICAL(i) = 'Index of oldest element in fifo queue p1-2' , 'Waiting to start production process one

and two'[p,i]) *aux, product unit, {Products, 'FIFO queue'}, zero order integration*

Start new production process three FOR(p=Products,i='FIFO queue' | IF('Ready to start new production process three' AND NUMERICAL(p) = 'Index of most urgent product p3' AND NUMERICAL(i) = 'Index of oldest element in fifo queue p3' , 'Waiting to start production process three'[p,i]) *aux, product unit, {Products, 'FIFO queue'}, zero order integration*

Start new production process two FOR(p=Products,i='FIFO queue' | IF(('Ready to start new production process two' AND NUMERICAL(p) = 'Index of most urgent product p1-2' AND NUMERICAL(i) = 'Index of oldest element in fifo queue p1-2')AND 'Process delay' <> TRUE , 'Waiting to start production process one and two'[p,i]) *aux, product unit, {Products, 'FIFO queue'}, zero order integration*

Start production process one and two TRANSPOSE({'Set one product each process one and two'}) *aux, product unit, {Products, 1...1}, zero order integration*

Start production process three TRANSPOSE({'Set one product each process three'}) *aux, product unit, {Products, 1...1}, zero order integration*

Stop for downtime p one 0 *const, unitless*

Stop for downtime p three 0 *const, unitless*

Stop for downtime p two 0 *const, unitless*

Sum fifo p1-2 ARRSUM('Waiting to start production process one and two',1,1) *aux, product unit, 'FIFO queue'*

Sum fifo p3 ARRSUM('Waiting to start production process three',1,1) *aux, product unit, 'FIFO queue'*

Sum waiting among oldest p1-2 IF(SIGN('Oldest products waiting p1-2') > 0, 'Sum waiting each product p1-2') *aux, product unit, Products*

Sum waiting among oldest p3 IF(SIGN('Oldest products waiting p3') > 0, 'Sum waiting each product p3') *aux, product unit, Products*

Sum waiting each product p1-2 ARRSUM('Waiting to start production process one and two',2,2) *aux, product unit, Products*

Sum waiting each product p3 ARRSUM('Waiting to start production process three',2,2) *aux, product unit, Products*

Waiting to start production process one and two 0«'product units'», inflow {FILLINZEROES('Start production process one and two')}, outflow {SUFFIXZERO('Shift fifo process one and two')}, inflow {PREFIXZERO('Shift fifo process one and two')} outflow {'Start new production process one'}, outflow {'Start new production process two'} *stock, product unit, {Products, 'FIFO queue'}*

Waiting to start production process three 0«'product units'», outflow {SUFFIXZERO('Shift fifo process three')}, inflow {PREFIXZERO('Shift fifo process three')}, inflow {FILLINZEROES('Start production process three')}, outflow {'Start new production process three'} *stock, product unit, {Products, 'FIFO queue'}*