The myth of speeding up business processes through parallel job processing

Michael Zapf, Ute Steidel, Armin Heinzl

Working Paper 9 / 2003
April 2003

Working Papers in Information Systems 1

University of Mannheim
Department of Information Systems 1
D-68131 Mannheim/Germany
Phone +49 621 1811691, Fax +49 621 1811692
E-Mail: wifo1@uni-mannheim.de
Internet: http://www.bwl.uni-mannheim.de/wifo1

The myth of speeding up business processes through parallel job processing

Michael Zapf[a], Ute Steidel[b] and Armin Heinzl[c]

Abstract. The current paper discusses the favorability of parallel job processing compared with the sequential design of business processes. A survey of relevant application domains, where parallelism of tasks is used to speed-up processes, shows two important points: paralleling of tasks can speed-up business processes, but an additional coordination effort may reduce or even invert the achievable performance gains. The relationship between this paralleling gain and coordination effort will be evaluated in detail through an extensive simulation study. The study examines paralleling patterns for a generic decision-making process within a public administration. Hereby, important knowledge about the favorability of parallel designs can be achieved: (a) paralleling gains are reduced with increasing process variability, (b) in some cases an additional coordination effort of 10-30% is enough for neutralizing the paralleling gains, (c) in high work load situations, the resource capacity for the coordination activity is a bottleneck for the overall processing speed and (d) the paralleling of multiple task leads to a higher performance gain.

Keywords. Business process design, coordination theory, paralleling patterns, decision-making processes, process simulation

---

[a] Michael Zapf, Information Systems 1, University of Mannheim, E-Mail: mzapf@uni-mannheim.de
[b] Ute Steidel, Accenture Germany, E-Mail: ute.steidel@accenture.com
[c] Armin Heinzl, Information Systems 1, University of Mannheim, E-Mail: aheinzl@uni-mannheim.de

1 Introduction

Business process management literature proposes parallel job processing as one important way to speed up business processes (e.g. Cuiper et al. 1996, Davenport 1993, Edosomwan 1996, Eversheim 1995, Goebel 1996, Hammer et al. 1993, Nissen 1998, Ould 1995, Schmelzer 1990, and Zangl 1985). General guidelines combined with success stories are used to deliver the substantial message: Parallelism of tasks reduce the overall throughput time! This message seems to be true at first sight but cannot be confirmed without restrictions on nearer view. Some authors mention that parallel process designs increase processing speed but may result in a higher coordination effort which accordingly reduces the expected efficiency gains (e.g. Achermann 1998, Adler et al. 1995, Davenport 1993, Hammer 1990, Zangl 1985). But there is no empirical evidence given for the superiority of parallel designs and no reasoning which helps to decide in which circumstances the gains of parallel job processing exceed the additional coordination effort.

In this paper we will provide a deeper analysis of the relation between performance gains through paralleling and the coordination effort. The influence of processing time variability and work load on the favorability of parallel process designs will be evaluated in detail.

The following study focuses on the deployment of multiple employees who perform tasks in parallel for the same job. In this case, additional coordination activities are necessary for reconciling the partial results of parallel tasks (see Adler et al. 1995, AitSahlia et al. 1995, Cuiper et al. 1996, Walter 1998). Another way of parallel job processing – which is not within the scope of this paper – would be the assignment of similar jobs to different resources who work in parallel on their tasks (see Blocher et al. 1996, Bukchin et al. 2003, Buzacott 1990, Buzacott 1996, Johnson 1983, Piersma et al. 1996, Pinto et al. 1975, Sheu et al. 1996 and Seidmann et al. 1997). In this classical problem class of queuing theory resources perform all tasks for one job and no intra-job coordination is necessary. The resource capacity is enhanced for a business process but not for a single job.

The paper is structured as follows: first the existing literature is reviewed in respect to application domains with parallel process designs. Empirical studies and analytical models are of special interest because they can give a clearer understanding of paralleling gains and restrictions. In this section also the relevant literature from coordination theory will be discussed in order to show the causes and effects of coordination activities within organizations.

After that, some generic paralleling scenarios will be developed for decision-making processes within public administrations. These scenarios are modeled as stochastic processing

networks and turn out to be useful for an exemplary study of the effects of parallel job processing and accompanying coordination activities. In the following section, model parameters, performance measures, and the employed evaluation technique are described and the numerical results of the simulation study are presented. Important managerial conclusions are derived at the end of the paper and future research directions are demonstrated.

2 Related contributions

Business process management

As outlined in the beginning, the business process management literature discusses parallel job processing superficially. General guidelines are given for designing business processes and the paralleling of tasks is regarded as one way to improve the efficiency of business processes regarding the overall throughput time. Some authors like Ould (1995, p. 157) give a simple reasoning for calculating the paralleling gain in a static environment: The paralleling of task A and B leads from a throughput time of $(t_A + t_B)$ in the sequential design to a throughput time of $max(t_A, t_B)$ in the parallel design, where t_i is the processing time of task i. Others may have this reasoning in mind when they just state parallel job processing as desirable and with that appeal to the intuitive understanding of their readers (e.g. Cuiper et al. 1996, Edosomwan 1996, Eversheim 1995, Goebel 1996, Nissen 1998, Schmelzer 1990 and Zangl 1985). Business cases are also provided to describe the advantages of parallel process designs but the basis and empirical data for the argumentation is not disclosed (e.g. Davenport 1993, Hammer et al. 1993).

The positive effects of parallel job processing are qualified in some contributions with the statement that parallel tasks cause a more complex process design and an increasing coordination effort (e.g. Achermann 1998, Adler et al. 1995, Davenport 1993, Hammer 1990, Zangl 1985). This additional effort reduces the process efficiency and requires more resources. One important factor which seems to reduce this effect is the utilization of information technology for the automation of coordination tasks (e.g. Davenport 1993, Hammer 1990, Walter 1998).

Product development and simultaneous engineering

An in-depth discussion of parallel job processing can be found in the product development domain, especially the discussions about simultaneous engineering. Within this approach, the product development time should be reduced through overlapping and paralleling of devel-

opment tasks (see Gerpott et al. 1996). Table 1 lists selected studies which examine the success of simultaneous engineering. The references are categorized concerning the applied method, scope and stated paralleling gains.

reference	method	scope	paralellization gains
AitSahlia et al. 1995	analytical model		positive, not quantified
Clark et al. 1991	empirical study (interviews)	20 companies from US, Western Europe and Japan, 29 projects (automobile)	positive, not quantified
Griffin 1993	empirical study (secondary)	21 companies mainly from US	-53% of development time
Handfield 1994	empirical study (interviews)	31 companies from US and Canada (job production)	-41% of development time, -49% of time to delivery for new products, +76% of time to delivery for developed products
Kessler et al. 1999	empirical study (interviews)	10 US companies, 75 projects	positive, not quantified
Murmann 1994	empirical study (interviews)	8 German companies, 14 projects (mechanical engineering)	-25% of development time
Schröder 1994	analytical model		positive, not quantified
Trygg 1993	empirical study (interviews)	109 Swedish companies (mechanical engineering and metal-processing)	-2% of development time
Wildemann 1993	empirical study (interviews)	12 German companies	from –30% to -50% of development time

Table 1. Studies regarding parallel job processing in the product development literature

Most of the listed publications state the effect of paralleling as positive regarding the reduction of development time. But the achievable gains are valued extremely different: Starting from 2% (Trygg 1993), gains go up to 53% (Griffin 1993). Some authors do not value the paralleling gains and Handfield (1994) mentions even a performance loss of 76% regarding the time to delivery for the enhancement of already developed products.

These results show that simultaneous engineering with a strong paralleling of development tasks may lead to performance gains but does not ensure the acceleration of the time to delivery. Hereby, especially the interpretation of the numerical results have to be made very carefully, because (a) the studies examine the influence of simultaneous engineering in its entirety

and not the paralleling of single tasks and (b) the studies base on interviews and therefore the quantification of paralleling gains cannot be analyzed in detail.

Parallel computing

Parallel job processing is the essential part of parallel computing. The execution of computer programs is accelerated through different types of parallelism in this domain which can be classified as follows (see Heiss 1994):

a) internal parallelism within one program (intra-program parallelism) and
b) external parallelism between multiple programs (inter-program parallelism).

The intra-program parallelism exploits the fact that computer programs contain not only parts with a sequential order but also parts which can be processed in parallel. The achievable performance gains depend on the number of parallel processors which are utilized for paralleling tasks and the communication time. Figure 1 shows the relation between the number of processors *n* and overall processing time *T(n)*. $T_x(n)$ is the execution time which is needed for executing the program instructions dependent on the number of processors *n*. It results from Amdahls law as monotonous decreasing curve since the achievable speed-up through paralleling is limited through sequential data dependencies which cannot be paralleled (see). As parts of a parallel program need interaction some coordination has to take place between the single parts. The accompanying coordination time $T_c(n)$ can be modeled as strict monotonous increasing curve with the number of processors (see). The overall processing time *T(n)* results as sum of execution and coordination time:

$$T(n) = T_x(n) + T_C(n) \quad .$$

The utilization of too many processors leads to an increase of processing time which is presented in Figure 1 for $n > n_{opt}$. This situation is called processor thrashing and expresses the fact that an increasing coordination effort occupies too many processors with unproductive work (see Heiss 1994). The trade-off between paralleling gains and communication effort results in an optimum number of processors n_{opt} with minimum processing time. In extreme cases the communication effort prohibits parallel processing, which has been shown in model calculations by Stone 1987.

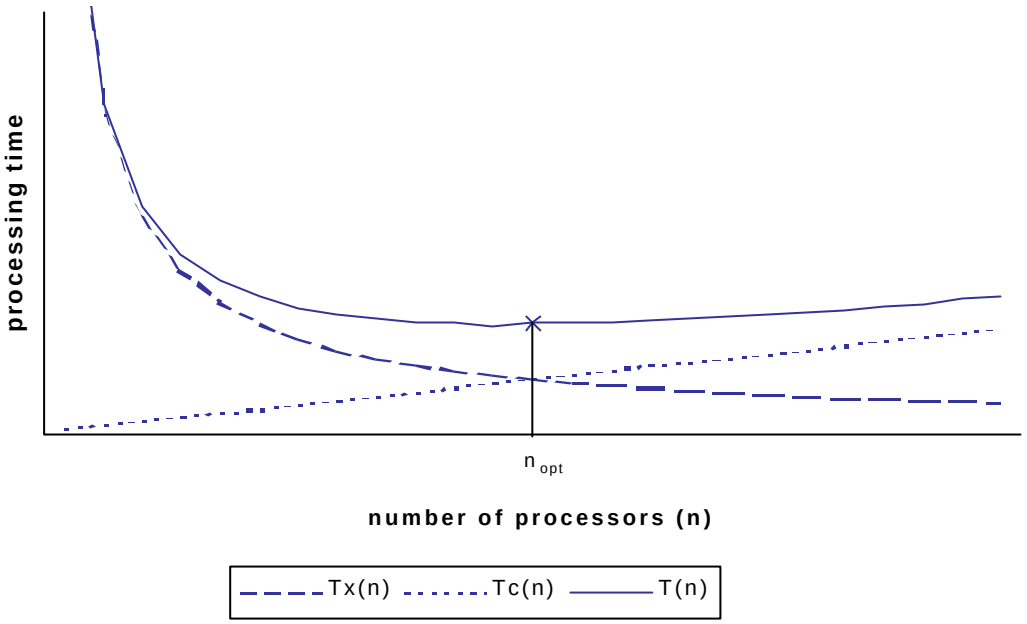

Fig. 1. Relation between number of processors and processing time for intra-program parallelism (see Heiss 1994, p. 51)

The inter-program parallelism deals with multiple programs which are executed in parallel on a shared amount of processors. The program instructions have to be distributed among the processors. According to the distribution strategy the paralleling gains of intra-program parallelism may be reduced in order to achieve a better utilization of processor resources (see Heiss 1994).

In summary the parallel computing literature states that parallel program execution leads to significant performance gains if (a) the assignment of different tasks on parallel processors is done efficiently and (b) no significant coordination effort has to be considered (see Heiss 1994, IEEE 1995, Seemers 1998). If parallel computing needs a costly intra-program or inter-program coordination the overall system has either to be designed very carefully (see Glücker 1998, Jaenicke 1998) or the performance gains may be limited or even reversed (see Hächler 1998, Smith 1999).

Coordination within organizations

The previous discussion of various application domains has shown that significant performance gains can be achieved through parallel job processing. But it has also been outlined that the performance of parallel designs depends strongly on the accompanying coordination effort. As the effects of coordination within organizations has been widely discussed in orga n-

izational literature, we will give a short overview of selected work from this field and transfer the statements to the application of parallel job processing.

Malone et al. (1994, p.90) defines coordination as "managing dependencies between activities". In the case of parallel business processes, dependencies may exist between paralleled tasks respectively their outcomes. Special activities are necessary in order to manage coordination, which Crowston (1997, p.169) calls coordination mechanisms. These additional activities point to additional effort within a process and are often called synchronization activities in parallel process designs. Malone et al. (1988, p.10) names the accompanying effort as coordination costs and uses the time as primary measure. He also assumes that coordination costs increase with the number of communication links. As parallel designs have more links than sequential designs, this would lead to the hypothesis that the coordination costs in parallel designs tend to be higher than the costs in a corresponding sequential design.

The connection between communication links (dependencies) and coordination costs goes back to the work of Thompson (1967). He differentiates between three types of dependencies within an organization: (a) pooled, (b) sequential and (c) reciprocal dependencies. The communication links and accompanying coordination effort increases from (a) to (c). This means for the performance of parallel designs that it is better if the dependencies within the process are pooled or sequential rather than being reciprocal. If the dependencies are reciprocal they should be performed with one organizational unit in order to avoid additional coordination costs between organizational boundaries. Kilman (1983) presents an approach for re-engineering the structure of an organization according to the task dependencies. The high coordination effort between different organizational units has also stated by Barua et al. (1996) who developed an analytical model for the information sharing between organizational units.

Housel et al. (1995) discusses the relation between process costs and process outcome. He suggests that a process should only consist of tasks which contribute value-added. On that score the question arises whether coordination activities contribute value-added and whether paralleling of tasks makes sense if additional coordination effort arises.

Pre-conclusion

The utilization of parallel job processing in different domains shows (a) that parallel designs can be more efficient than sequential designs and (b) the necessary coordination activities play an important role for the favorability of parallel designs. This relationship will be ana-

lyzed in the following with the help of a generic decision-making process within a public administrative office.

3 Alternative paralleling scenarios for a generic decision-making process

Three different scenarios are used for evaluating the effects of parallel job processing. Each scenario refers to a typical decision-making process within a public administration office, which is, for example, utilized for the handling of building requests.

The first scenario describes the sequential processing of a single request. It is used as reference base for the analysis of the following parallel processing models. The according business processes is presented as Petri Net in Figure 2.

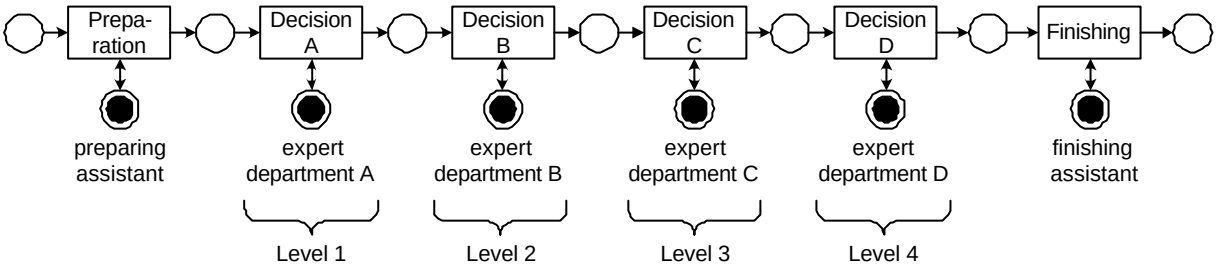

Fig. 2. Scenario I: Sequential decision-making process with 4 levels

First the incoming documents are accepted by a preparing assistant. He performs a completeness check, collates the documents and forwards them to the first expert of department A. The expert evaluates the request and draws up an expert opinion which is attached to the record. After that the record is passed on to an expert of department B and so on. At the end of the valuation process the expert opinions are summarized by a finishing assistant who prepares the record for the decision committee who makes a final decision about the acceptance or refusal of the request.

Two other designs will be derived from this base scenario with different degrees of parallelism. The following premises have to be kept in mind for the resulting models:

- Processing order: we assume that decision C is dependent on decision A. Moreover, decision D is dependent on B. Therefore A has to be performed before C and in addition B has to be performed before D. No other chronological dependencies exist within the overall process.

- One activity is performed by one person. No additional coordination takes place between the departments.
- One person performs one activity at a certain point in time and does not evaluate multiple requests simultaneous.
- Requests are handled according to the first in first out (FIFO) strategy. Activities which have started once will be not interrupted by other requests.

In the first step, we derive Scenario II which emerges from scenario I through paralleling decisions A and B, see Figure 3. The preparing assistant duplicates all documents and forwards them in parallel to expert A and expert B. Both experts pass on their opinion to expert C after finishing the evaluation. Expert C starts his task after receiving both expert opinions. The further processing remains the same as in scenario I. With this scenario two positive effects are expected: (a) the shortening of the processing time and (b) a better decision quality because of the independence of expert A and expert B.

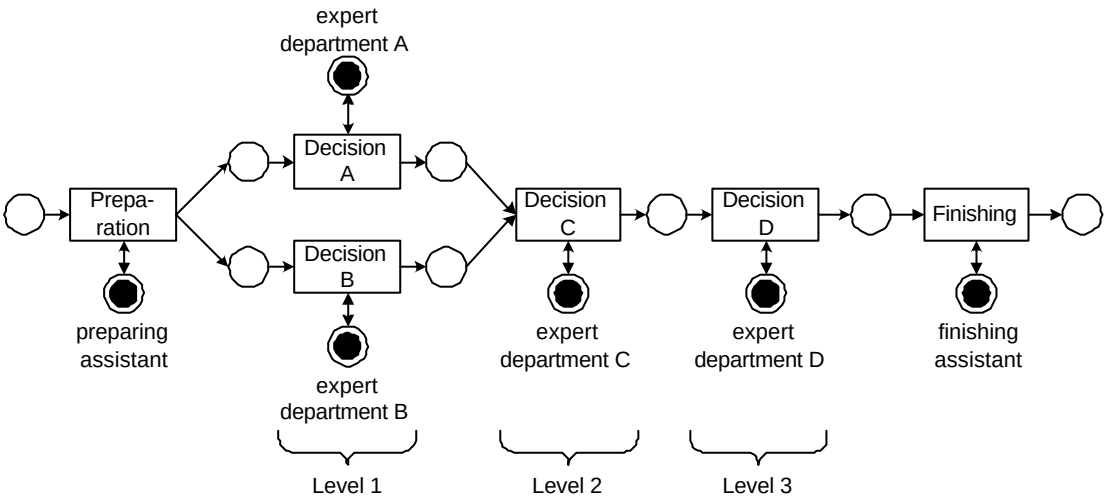

Fig. 3. Scenario II: Parallel decision-making process with 3 levels

The third scenario III implies further parallelism of activities: Decisions A and C are performed in parallel to decisions B and D. This leads to a process with two levels as presented in Figure 4. Expert A forwards his results to expert C and expert B forwards his results to expert D. The valuation reports are summarized by the finishing assistant. The decision committee weighs up the different expert opinions and comes to a final decision.

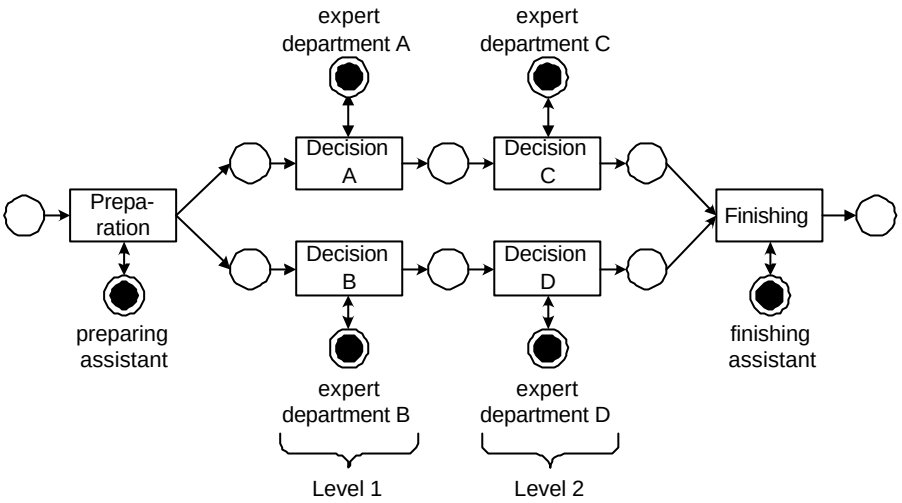

Fig. 4. Scenario III: Parallel decision-making process with 2 levels

4 Methodology

We use an experimental approach based on computer simulation for evaluating the paralleling scenarios described above. This kind of research design has been widely and successfully used for evaluating business process designs (e.g. Adler et al. 1995, Choi et al. 1998, De Vreede et al. 1996, Farrington et al. 1998, Giannini et al. 1997, Prietula et al. 1994, Shanker et al. 1985, Zapf et al. 2000, Zapf 2001, Zapf et al. 2001).

As the paralleling scenarios are non-static discrete systems they will be represented as discrete event computer models and analyzed with the help of stochastic simulation (see Law et al. 1991). For preparing the simulation model and performing the experiments the simulation tool ARENA is used (Kelton et al. 1998). The single experiments are performed in the form of a steady-state simulation in order to analyze the long-term effect of parallelism. Every experiment consists of 1.000 independent runs. This number is much higher than general rules from literature and ensures the validity of the obtained results[1]. Every run represents 2.000 hours of work which corresponds approximately to one working year (50 weeks, 5 days per week, 8 hours per day). The warm-up period of 3.000 hours has not been included in the results.

[1] Bulgren, 1982 suggests for example 30 independent runs.

5 Model parameters and performance measures

An overview of the model parameters is presented in Table 2. The inter-arrival time gives the time between the arrival of two different requests. As in most practical applications the inter-arrival time is extremely variable we model it as stochastic parameter and use the exponential distribution, which has been frequently proposed for incoming customer requests in literature (see Kelton et al. 1998, Liebl 1995, Sheu et al. 1996, Buzacott 1996, Seidmann et al. 1997). Only in the first experimental series, it is assumed as deterministic parameter in order to get a reference base for the experimental study.

parameter	**description**	**characteristic**
inter-arrival time	time between the arrival of two different requests	deterministic, stochastic
handling time	time for executing a task for one request	deterministic, stochastic
coordination factor	models the additional effort for the synchronization of parallel tasks, the factor is multiplied with the processing time of the synchronization activity	deterministic

Table 2. Model parameters

The handling time is needed for executing one single task. It is dependent on the task type (preparation, decision or finishing task) and the difficulty of the current request. In order to evaluate the influence of the handling time on the paralleling gains we model the handling time alternatively as (a) deterministic and (b) stochastic parameter with triangular distribution. Two different experimental series are separated in the stochastic case: (b1) balanced system and (b2) imbalanced system. The handling times of all tasks within the process have the same average value within a balanced system. Within imbalanced systems, the average handling times vary for different tasks.

The coordination factor is the central parameter for our analysis. This parameter is introduced in order to reflect the coordination costs which are caused by an additional effort for coordinating parallel tasks. The explicit reflection of coordination costs within our analysis goes back to the work of Thompson (1967), Kilman (1983) and Malone et al. (1988) who explain the effects of coordination effort between organizational units and show ways to reduce it (see paragraph 2).

The coordination factor is modeled as percentage of the handling time. A high coordination effort (e.g. 180%) leads to a higher average handling time (e.g. 180% of the original handling

time). If an expert opinion is passed through a sequential process, for example, the single experts add some additional points or deny some arguments of their predecessor(s). The structure of the first expert opinion is normally taken over and therefore the summarization at the end of the process is not too complicated. Every expert has to generate his own arguments and statements in the case of parallel decision. Synchronizing these different arguments may be more difficult and time-consuming than in the sequential scenario. We assume a proportional coordination effect in relation to the average handling time and therefore use a factor for representing the additional coordination effort.

This study evaluates the influence of paralleling on the processing speed and therefore uses the throughput time as the main performance measure. The throughput time begins with the arrival of a new request and ends after completing the finishing activity. It consists of handling time and turnaround time. Within the experiments we measure the average processing time over all requests.

The availability of resources is crucial for the process performance. The throughput time, for example, increases exponentially dependent on the utilization of employees (see Kleinrock 1975). In order to evaluate paralleling gains in relation to the work load of employees, the average utilization of employees is measured additional.

6 Numerical analysis

The deterministic starting point

The numerical analysis is started with a simple reasoning in order to take up the general statements of many authors, who propose parallel job processing for speeding up business processes without any restrictions (see section 1). Therefore, a static environment is assumed with a constant processing time of 11:40 hours per task, a constant arrival rate and no turnaround time. This calculation gives the maximum gain which can be achieved through parallel job processing.

Without any additional coordination effort (k = 100) the difference between scenario I and scenario II is 11:40 hours which is exactly the processing time of one task (see Figure 5). The break-even point for scenarios I and II is k = 200. This means that the parallel job processing is better than the sequential processing as long as the additional coordination effort does not exceed the processing time of one complete task. The additional parallelism in scenario III improves the process performance furthermore and leads to an efficiency gain of 23:20 hours

in the best case (k = 100). The break-even point for scenario I and III is k = 300, which is out of the scope of Figure 5. These simple calculations seem to prove that parallel job processing may lead to a maximum speed-up of 17% for scenario II (= 11:40 hours) and 33% for scenario III (= 23:20 hours).

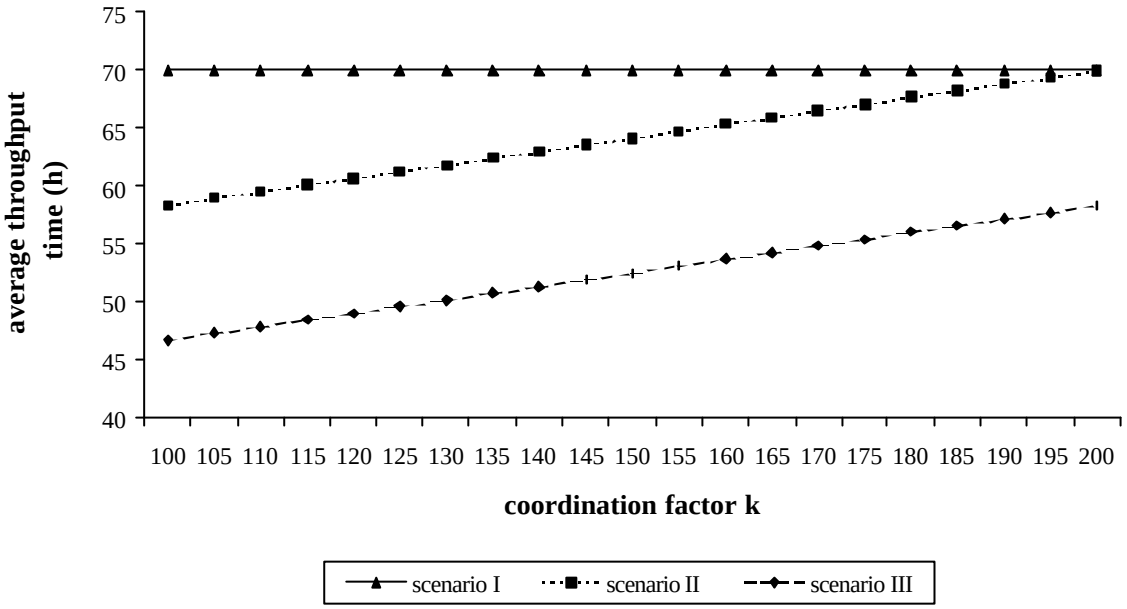

Fig. 5. Average throughput time for scenarios I–III in a deterministic environment with deterministic work load and processing time

Experimental Design

The assumption of constant arrival rate and processing rate does not hold true in most practical environments (see Kelton et al. 1998). Therefore, we perform multiple experiments with different levels of variability regarding the inter-arrival time and the processing time. The analyzed variability levels and parameter constellations are listed in Table 3.

The inter-arrival time is modeled with the exponential distribution which is often used in comparable experimental studies (see Sheu et al. 1996, Buzacott 1996, Seidmann et al. 1997). The utilized distributions for three different workload situations low, medium and high are shown in Table 3 with the corresponding mean values. Expo(45) is used as an abbreviation for an exponential distributed inter-arrival time with a mean value of 45 hours. In the deterministic environment the inter-arrival time is not exactly specified, because it has no influence on calculating the throughput time as long as it is greater than the processing time of one task (11:40 hours).

The triangular distribution is utilized within the stochastic environments for the processing time (see Kelton et al. 1998). For every triangular distribution the minimum, mode and maximum values is given in Table 3 and abbreviated as tria(minimum, mode, maximum). At this point we emphasize that the mean value of the triangular(3, 8, 24)-distribution calculates as (3 + 8 + 24)/3 h = 11:40h and with that corresponds to the deterministic environment. Within the stochastic imbalanced environment, decision A has a shorter and decision B a longer processing time (see Table 3). The corresponding mean values 5:40h and 17:40h sum up to 23:20h which is twice as much as the deterministic processing time. So the sum of the processing time mean values remains the same over all environments.

experimental series	**inter-arrival time**	**processing time**		
		deter-ministic	**stochastic balanced**	**stochastic imbalanced**
deterministic environment	> 11:40 h	11:40 h	./.	./.
processing variability, low work load	expo(45) h	11:40 h	tria(3,8,24) h	decision A: tria(1,4,12) h decision B: tria(5,12,36) h others: tria(3,8,24) h
processing variability, medium work load	expo(25) h	11:40 h	tria(3,8,24) h	decision A: tria(1,4,12) h decision B: tria(5,12,36) h others: tria(3,8,24) h
parallelism scope, medium work load	expo(25) h	./.	tria(3,8,24) h	./.
parallelism scope, high work load	expo(14) h	./.	tria(3,8,24) h	./.

Table 3. Experimental design and parameter values[2]

The influence of processing time variability

In this section, we include the influence of processing time variability in our break-even analysis. The sequential reference scenario I is compared to scenario II which realizes the parallelism of two tasks and comprises 3 decision levels.

The requests arrive with an average inter-arrival time of 45 hours in the “low work load” environment which leads to a low amount of work for the employees. The maximum average utilization of employees lies between 39,04% and 41,32%. Figure 6 presents the differences between scenario I and scenario II in respect to the average throughput time for different co-

[2] Legend: expo = exponential distribution, tria = triangular distribution.

ordination factors. A positive value indicates that parallel job processing is superior to the sequential design.

With deterministic processing time and stochastic arrival rate the break-even point is k = 160 (deterministic). This shows a substantial difference to the starting point with deterministic arrival rates where the break-even point was k = 200 (see Figure 7) and indicates that the advantages of parallelism decreases with the variability of the inter-arrival time.

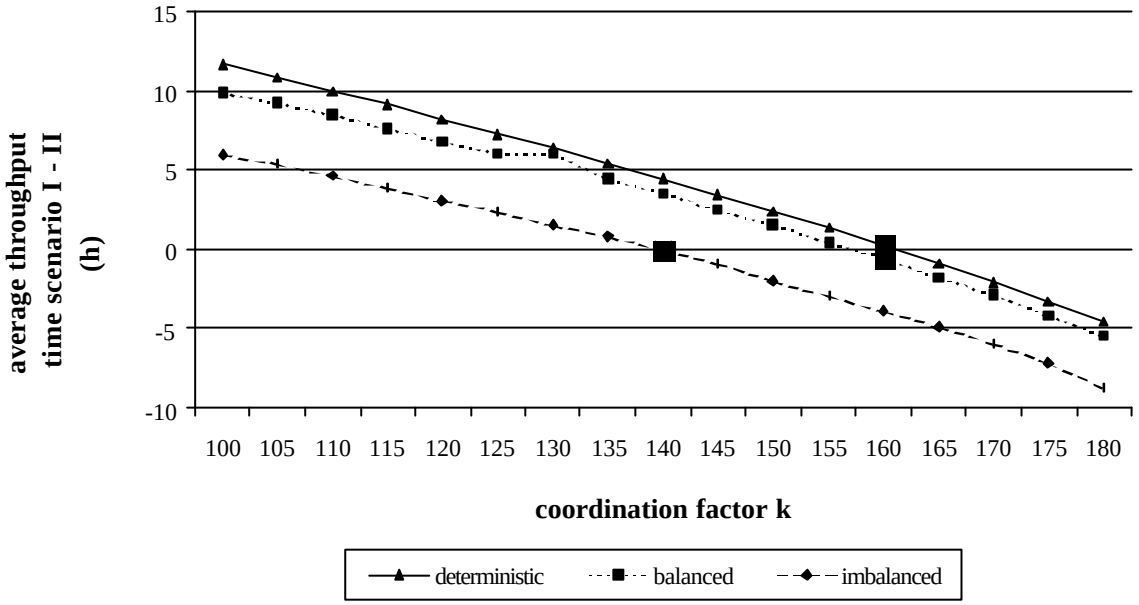

Fig. 6. Differences between the average throughput time of scenario I and II for different variability levels in a low work load environment

The introduction of further variability into the process leads to even less efficiency of scenario II. In the balanced environment with equal stochastic processing time per task the performance of scenario II decreases a bit but the break-even point remains k =160. Imbalanced tasks make the situation even worse. Parallel job processing is inferior for coordination factors k > 140 in this case.

The performance losses of scenario II with increasing variability can also be observed in the "medium work load" environment (see Figure 7). This environment is characterized by an average inter-arrival time of 25 hours which results in a maximum average utilization over all involved persons between 62,87% and 70,43%. In the deterministic and stochastic balanced environments the parallelism scenario II is superior for k < 135. In the case of stochastic imbalanced processing time the break-even point goes down to k = 130.

The experiments of this section show that a parallel process design performs better than a sequential design. But the advantage decreases with an increasing level of processing time

variability and increasing work load. These findings can be explained by the existence of an additional turnaround time in scenario II, which comes up if decision A takes longer than decision B or vice versa. The process continues not before both tasks have been finished and therefore always the longest processing time determines the overall throughput time.

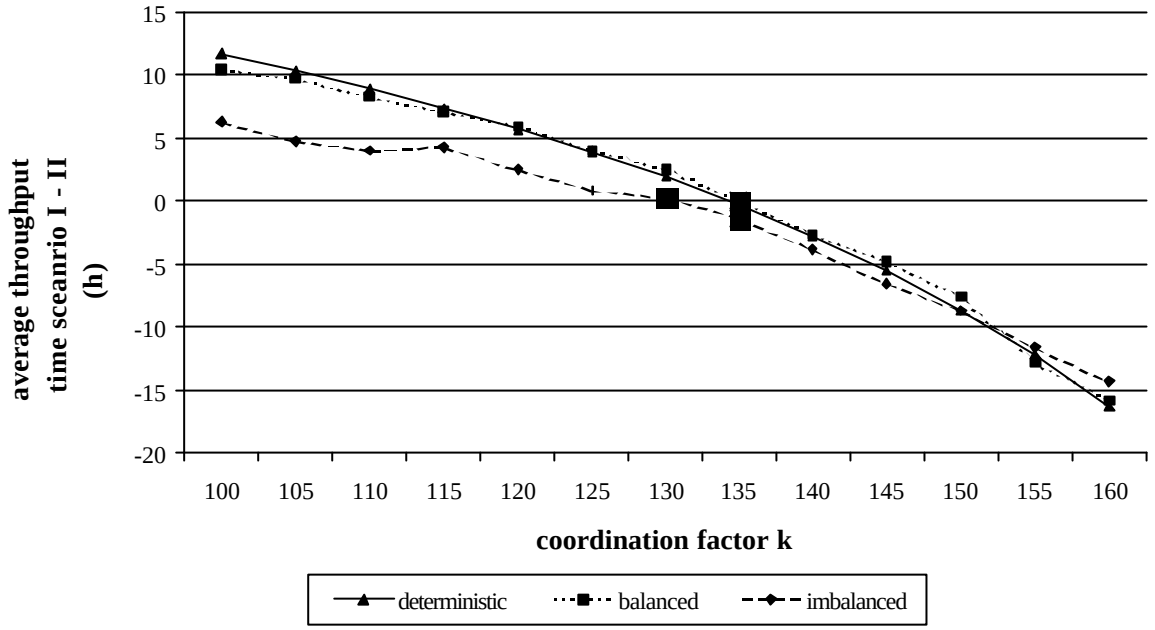

Fig. 7. Differences between the average throughput time of scenario I and II for different variability levels in a medium work load environment

If an additional coordination effort is necessary to synchronize the parallel tasks, the parallel processing may even be worse than the sequential design. Dependent on the processing time variability and the work load the break-even coordination factor k lies between 130 and 160. Thus, in some cases an additional coordination effort of 30% makes the sequential job processing recommendable.

The parallelism scope and achievable efficiency gains

In the previous section we examined the performance gains through paralleling two tasks with different levels of processing time variability. In this section, different parallelism scopes are compared with each other in the stochastic balanced environment. First the comparison is based on a medium work load with an average inter-arrival time of 25 hours and a resulting maximum average utilization between 62,56% and 72,14%.

Figure 8 shows that the more extensive parallelism in scenario III reduces the average throughput time and moves the break-even point from k = 135 (scenario II) to k = 155 (sce-

nario III). This indicates the processing time variability can be better compensated in scenario III than in scenario II.

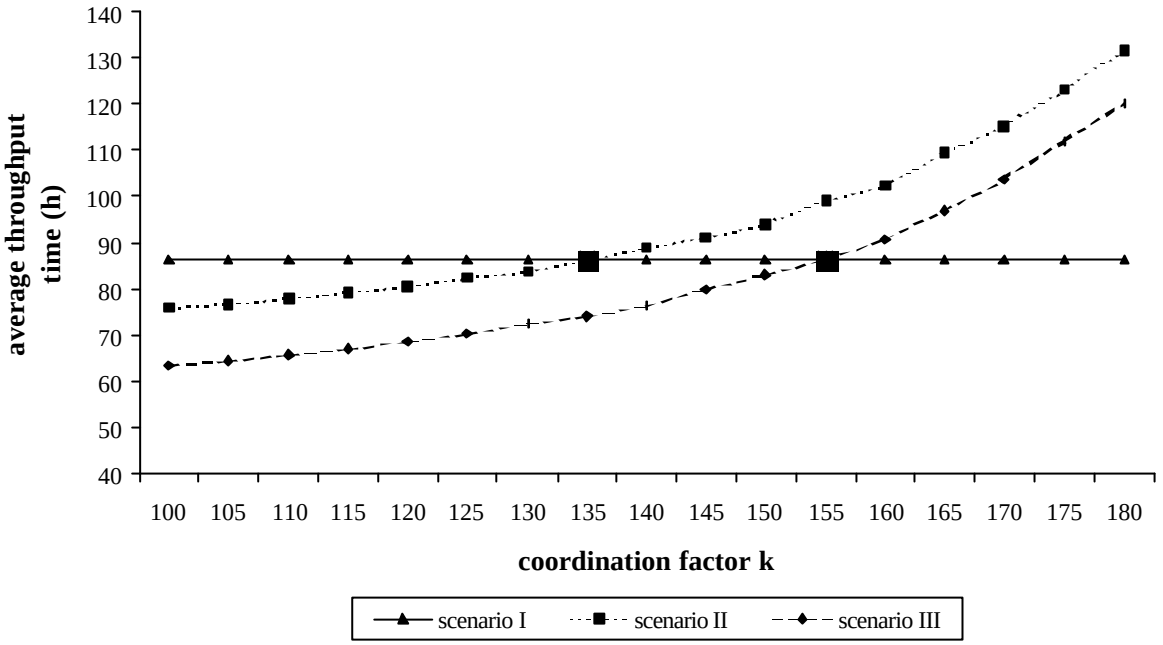

Fig. 8. Average throughput time for scenarios I–III with stochastic balanced processing time in a medium work load environment

The second experimental series is performed under a high work load environment with an average inter-arrival time of 14 hours. The resulting maximum average utilization of employees lies between 87,21% and 91,63% which reflects an overload situation.

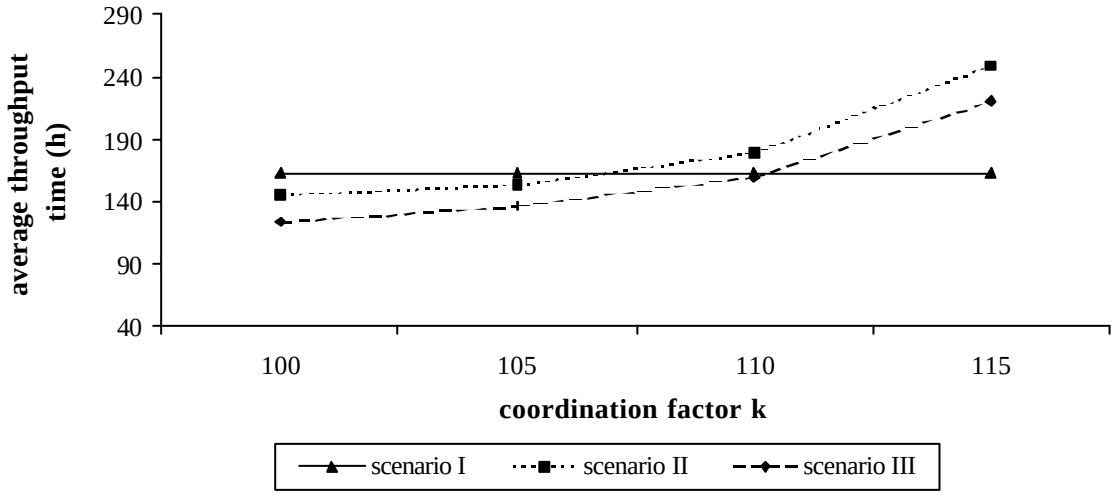

Fig. 9. Average throughput time for scenarios I–III with stochastic balanced processing time in a high work load environment

Figure 9 shows that an additional coordination effort has strong influence on the favorability of the parallel designs. A coordination effort of 10% or more makes the parallel job processing inferior to the sequential design. This can be explained with the extreme utilization of employees. Little increases of the processing time lead to disproportionate increases in the throughput time.

7 Discussion

Through the previous numerical analysis of typical paralleling patterns, we achieved important knowledge about using parallel tasks as means for (re-)designing business processes. The following findings have to be taken into consideration when parallel job processing should be used for speeding-up business processes.

Variability reduces paralleling gains

In dynamic environments the variability of inter-arrival and processing time lead to an additional turnaround time. If job B arrives shortly after job A, B has to wait in the queue until the previous job A has been finished. Or if the processing of job A takes longer than expected, the next job B, which has already arrived, has to wait in the queue. This turnaround time occurs in both sequential and parallel designs and leads to an increasing average throughput time.

An additional turnaround time exists in parallel designs for two parallel tasks A and B. Even if both tasks have the same *average* processing time this does not mean that they have exactly the same processing time for one specific job. If task A has finished but task B has not, the job cannot be processed until task B has also finished. This holds true whether the succeeding employee is free or busy. So for jobs an additional turnaround time and for employees an additional idle time is introduced with parallelism.

These effects lead to the observation, that the performance gains through parallelism are lower in dynamic environments than in static environments. Thus, it is important for the process designer to analyze the variability of the environment in detail. Important influence factors are the inter-arrival time, the processing time and the balance of tasks. With increasing variability and task imbalance the favorability of parallel designs decrease.

Additional coordination effort slows down job processing more than may be expected

The performance gains of parallel designs are strongly dependent on the coordination effort which is necessary for synchronizing the results of parallel tasks. Within a break-even analysis we determined the coordination factor which compensates the paralleling gains. The break-even point lies between 110% and 160% of the original synchronization time and depends on the work load and the variability of the environment. Higher work load and higher variability lead to a lower break-even point which reflects a poorer performance of parallel designs. This effect can be explained with the excessive utilization of the synchronizing employee in the case of an additional coordination effort. This employee performs the coordination activities and gets the bottle-neck for the overall process.

Thus, the existence and strength of an additional coordination effort has to be estimated before (re-)engineering the process. One way to make parallel designs more favorable would be providing additional resources for coordination activities. Another way would be automating coordination so that no or few additional coordination activities have to be performed manually.

Coordination activities may be bottle-necks

The utilization of every involved employee is important for the overall process performance. If one employee is in an overload situation, he is a bottle-neck and the whole processing speed breaks down. As described above this is especially true for the employee who performs the coordination activity. In our experiments with high work loads we found out that a resource utilization of more than 80% leads to extreme turnaround times and a noticeable slow-down of the overall process.

From the performance perspective it is important to avoid bottle-necks within the process and provide enough resources especially for performing the coordination task. On the other hand do additional coordination resources increase the overall process costs. This trade-off is in line with the results of organizational literature (see paragraph 2) and has to be clear to the process designer in order to make the right choice.

Paralleling task sequences is more efficient than paralleling single task

The parallelism scope has strong influence on the overall processing speed. The performance gains are higher if sequences of tasks are paralleled in contrast to parallel single tasks. We

examined this effect regarding the paralleling of two-task-sequences versus paralleling single tasks. The two-task-sequences lead to a "shortening" of the process through an additional parallel processing while still maintaining only one synchronization point. Thus, additional parallelism can be introduced while the coordination effort remains the same as in the case of paralleling single tasks. In this context we have to point out that this holds only true if the paralleled task-sequences are balanced, which means that they have the same average processing time. Unbalanced task-sequences will reduce the additional paralleling gain.

If parallel job processing is considered for a business process not only single tasks but longer task-sequences should be paralleled.

8 Summary and further research

This paper discusses the favorability of parallel job processing compared with the sequential design of business processes. A survey of relevant application domains, where parallelism of tasks is used to speed-up processes, has shown two important points: (a) the paralleling of tasks can speed-up business process, but (b) an additional coordination effort may reduce or even invert the achievable performance gains.

The relationship between paralleling gain and coordination effort has been evaluated in detail through an extensive simulation study of generic paralleling patterns. Hereby, important knowledge about the favorability of parallel designs could be achieved: (a) paralleling gains are reduced with increasing process variability, (b) in some cases an additional coordination effort of 10-30% is enough for neutralizing paralleling gains, (c) the resource capacity for the coordination activity is a bottleneck for the overall processing speed in high work load situations and (d) the paralleling of multiple task leads to a higher performance gain.

Our analysis has shown some important effects within parallel process designs. In order to confirm and enhance our results for other designs and environments, further research work is suggested. The analysis could be enhanced according to the following dimensions:

- Instead of using one employee per task, multiple employees are employed for one task.
- Besides from using the exponential and triangular distribution, other theoretical and empirical stochastic distributions have to be tested.
- Additional to the throughput time other effects e.g. regarding the quality of the process outcome have to be evaluated.
- Concerning the coordination effort we focused on the synchronization point. Further work has to examine additional coordination activities during the execution of the parallel tasks.

- The coordination effort has to be categorized and quantified within empirical studies in different application domains and for different business processes in order to get a clearer understanding of the coordination volume in special environments.

The presented experimental analysis is not intended to be a comprehensive treatment of parallel job processing within business processes but rather a starting point for further work in this area. Our initial experiments have shown the importance and effects of coordination effort within parallel process designs. This knowledge should be exploited when business processes are (re)designed within organizations.

9 References

Achermann B (1998) *Aufbau eines Outbound Telesales Centers in 12 Wochen*. HMD 35 (204): 45-55.

Adler PS, Mandelbaum A, Nguyen V, Schwerer E (1995) *From project to process management: An empirically-based framework for analyzing product development time*. Management Science 41 (3): 458-484.

AitSahlia F, Johnson E, Will P (1995) *Is concurrent engineering always a sensible proposition?* IEEE Transactions on Engineering Management 42: 166-170.

Amdahl G (1967) *Validity of the single processor approach to achieving large scale computer capabilities*. Proceedings AFIPS Comp. Conference 30: 483.

Barua A, Ravindran S (1996) *Reengineering information sharing behaviour in organizations.* Journal of Information Technology 11: 261-272.

Blocher JD (1996) *The customer order lead-time problem on parallel machines*. Naval Research Logistics 43: 629-654.

Bukchin J, Rubinovitz J (2003) *A weighted approach for assembly line design with station paralleling and equipment selection.* IIE Transactions 35: 73-85.

Bulgren WG (1982) *Discrete System Simulation,* Englewood Cliffs.

Buzacott JA (1990) *Abandoning the moving assembly line: models of human operators and job sequencing.* International Journal of Production Research 28 (5): 821-839.

Buzacott JA (1996) *Commonalities in Reengineered Business Processes: Models and Issues.* Management Science 42 (5): 768-782.

Choi S-H, Kin J-S (1998) *A study on the measurement of comprehensive flexibility in manufacturing systems*. Computers and Engineering 34 (1): 103-118.

Clark KB, Fujimoto T (1991) *Product development performance,* Boston.

Crowston KA (1997) *Coordination Theory Approach to Organizational Process Design.* Organization Science 8 (2): 157-175.

Cuiper R, Feldmann C, Rossgoderer U (1996) *Rechnerunterstützte Parallelisierung von Konstruktion und Montageplanung*. Zeitschrift für wirtschaftlichen Fabrikbetrieb 91 (7): 338-341.

Davenport TH (1993) *Process Innovation: Reengineering work through information technology,* Boston.

De Vreede GJ, Van Eijck DTT, Sol HG (1996) *Dynamic modelling for re-engineering organizations.* INFOR 34 (1): 28-42.

Edosomwan JA (1996) *Organizational transformation and process reengineering,* Delray Beach, Florida.

Eversheim W (1995) *Prozessorientierte Unternehmensorganisation,* Berlin et al.

Farrington P, Nazemetz JW (1998) *Evaluation of the performance domain of cellular and functional layouts*. Computers and Engineering 34 (1): 91-101.

Gerpott TJ, Winzer P (1996) *Simultaneous Engineering: Kritische Analyse eines Planungs- und Organisationsansatzes zur Erfolgsverbesserung industrieller Produktinnovationen.* Zeitschrift für Planung (2): 132-150.
Giannini PJ, Gruppe FH, Saholsky RM (1997) *Reengineering through simulation modeling: Optimizing a telephone ordering system at GPO*. Information Systems Management 14 (3): 61-66.
Glücher R (1998) *Durchsatzsteigerung in einem Image-Processing-System durch Parallelverarbeitung*. HMD 35 (203): 58ff.
Goebel E (1996) *Prozessorganisation - radikaler Neubeginn oder alte Wissensbestände im neuen Gewande.* Zeitschrift für Planung (4): 309-318.
Griffin A (1993) *Metrics for measuring product development cycle time*. Journal of Product Innovation Management 10 (1993): 112-125.
Hächler G (1998) *Die parallele Koordinationssprache ALWAN.* HMD 35 (203): 38ff.
Hammer M, Champy J (1993) *Reengineering the Corporation: A Manifesto for Business Revolution,* New York.
Hammer M (1990) *Reengineering work: Don't automate, obliterate.* Harvard Business Review July-August: 104-112.
Handfield RB (1994) *Effects of concurrent engineering on make-to-order products.* IEEE Transactions on Engineering Management 41: 384-393.
Heiss H-U (1994) *Prozessorzuteilung in Parallelrechnern,* Mannheim et al.
Housel T, Kanevsky VA (1995) *Reengineering business processes: A complexity theory approach to value added.* INFOR 33 (4).
IEEE (1995) *Special Issue on Microprocessors.* Proceedings of the IEEE 83 (12).
Jaenicke R (1998) *Parallel processing: The future of embedded systems.* Canadian Electronics 13 (3).
Johnson RV (1983) *A branch and bound algorithm for assembly line balancing problems with formulation irregularities*. Management Science 29 (11): 1309-1324.
Kelton WD, Sadowski RP, Sadowski DA (1998) *Simulation with ARENA,* Boston et al.
Kessler E, Chakrabarti AK (1999) *Speeding up the pace of new product development.* Journal of Product Innovation Management 16 (3): 231-247.
Kilman RH (1983) *The costs of organization structure: Dispelling the myths of independent divisions and organization-wide decision making.* Accounting, Organizations and Society 8 (4): 341-357.
Kleinrock L (1975) *Queuing systems,* New York et al.
Law AM, Kelton WD (1991) *Simulation modeling & analysis,* New York et al.
Liebl F (1995) *Simulation*, München et al.
Malone TW, Smith SA (1988) *Modeling the performance of organizational structures.* Operations Research 36 (3): 421-436.
Malone TW, Crowston K (1994) *The interdisciplinary study of coordination.* ACM Computing Surveys 26 (1): 88-119.
Murmann PA (1994) *Expected development time reductions in the German mechanical engineering industry*. Journal of Product Innovation Management 11: 236-252.
Nissen ME (1998) *Redesigning reengineering through measurement-driven inference.* MIS Quarterly December: 509-534.
Ould M (1995) *Business processes: Modelling and analysis for re-engineering and improvement,* Chichester et al.
Piersma N, Romeijn HE (1996) *Parallel machine scheduling: a probabilistic analysis*. Naval Research Logistics 43: 897-916.
Pinto PA, Dannenbring DG, Khumawala BM (1975) *A branch and bound algorithm for assembly line balancing with paralleling.* International Journal of Production Research 13 (2): 183-196.
Prietula MJ, Carley KM (1994) *Computational organization theory: autonomous agents and emergent behavior*. Journal of Organizational Computing 4 (1): 41-83.
Schmelzer HJ (1990) *Steigerung der Effektivität und Effizienz durch Verkürzung von Entwicklungszeiten.* In: Reichwald R, Schmelzer HJ (eds) Durchlaufzeiten in der Entwicklung: Praxis des industriellen F&E Managements, München, pp 27-64.
Schröder H-H (1994) *Die Parallelisierung von Forschungs- und Entwicklungs-(F&E)-Aktivitäten als Instrument zur Verkürzung der Projektdauer im Lichte des "Magischen Dreiecks" aus Projektdauer,*

Projektkosten und Projektergebnissen. In: Zahn E (ed) Technologiemanagement und Technologien für das Management, pp 289-323.
Seidmann A, Sundararajan A (1997) *The Effects of Task and Information Asymmetry on Business Process Redesign.* International Journal of Production Economics 50 (2-3): 117-128.
Shanker K, Tzen Y-JJ (1985) *A loading and dispatching problem in a random flexible manufacturing system.* International Journal of Production Research 23 (3): 579-595.
Sheu C, Babbar S (1996) *A managerial assessment of the waiting-time performance for alternative service process designs.* Omega 24 (6): 689-703.
Siemers C (1998) *Paralele Programmierung - Nicht ohne Prozessor- und Rechnertechnik.* HMD 35 (203): 9-20.
Smith B (1999) *Breaking through the I/O bottleneck.* Network World 16 (34).
Stone HS (1987) *High performance computing architecture,* Reading Mass.
Thompson JD (1967) *Organizations in Action: Social Science Bases of Administrative Theory,* New York et al..
Trygg L (1993) *Concurrent Engineering practices in selected Swedish companies: A movement or an activity of the few?* Journal of Production Innovation Management 10: 403-415.
Walter ZD (1998) *Workflow management in business processes,* Dissertation University of Rochester, New York.
Wildemann H (1993) *Just-In-Time in Forschung & Entwicklung und Konstruktion.* Zeitschrift für Betriebswirtschaft 63: 1251-1270.
Zangl H (1985) *Durchlaufzeiten im Büro,* Berlin.
Zapf M, Heinzl A (2000) *Evaluation of Generic Process Design Patterns: An Experimental Study.* In: van der Aalst WMP, Desel J, Oberweis A (eds) Business Process Management: Models, Techniques, and Empirical Studies, LNCS 1806, Berlin et al.
Zapf M (2001) *Gestaltung flexibler Kundeninteraktionsprozesse im Communication Center: Theoretische Grundlagen und experimentelle Analyse,* Dissertation, University of Bayreuth.
Zapf M, Storch K (2001) *Making Simulation Work for the Organizational Design of Communication Centers: Challenges and Practical Experience.* Proceedings of the 2001 Summer Computer Simulation Conference, Orlando.