View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Research Papers in Economics

Concurrent Engineering
http://cer.sagepub.com

Collaborative Metamodeling: Coordinating Simulation-based Product Design

Bart Husslage, Edwin van Dam, Dick den Hertog, Peter Stehouwer and Erwin Stinstra

Concurrent Engineering 2003; 11; 267
DOI: 10.1177/1063293X03039895

The online version of this article can be found at:
http://cer.sagepub.com/cgi/content/abstract/11/4/267

Published by:
SAGE Publications
http://www.sagepublications.com

Additional services and information for *Concurrent Engineering* can be found at:

Email Alerts: http://cer.sagepub.com/cgi/alerts

Subscriptions: http://cer.sagepub.com/subscriptions

Reprints: http://www.sagepub.com/journalsReprints.nav

Permissions: http://www.sagepub.com/journalsPermissions.nav

Downloaded from http://cer.sagepub.com at Universiteit van Tilburg on May 14, 2007
© 2003 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

CE CONCURRENT ENGINEERING: Research and Applications

Collaborative Metamodeling: Coordinating Simulation-based Product Design

Bart Husslage,[1,*] Edwin van Dam,[1] Dick den Hertog,[1] Peter Stehouwer[2] and Erwin Stinstra[2]

[1]*Department of Econometrics and Operations Research, Tilburg University, P.O. Box 90153, 5000 LE Tilburg, The Netherlands*

[2]*Centre for Quantitative Methods, P.O. Box 414, 5600 AK Eindhoven, The Netherlands*

Abstract: High-tech products, like automobiles and aircrafts, consist of many components. To evaluate component designs, simulation tools are frequently used. However, component complexity often results in very large simulation times, so the number of evaluated designs is limited. Further, relations among components make the design of the final product a very hard task that can only be accomplished by a proper coordination of all the simulation tools.

The first part of this paper gives a framework to deal with simulation-based product design for cases where there are multiple coupled simulation tools and large simulation times. We call our approach Collaborative Metamodeling (CMM). The CMM approach focuses on the construction of metamodels for components, in order to gain insight in the behavior of components and the final product. This knowledge is used in finding an optimal and robust product design.

The second part of this paper deals with the coordination of simulation tools. Therefore, we define three coordination methods: Parallel simulation, Sequential simulation, and Sequential modeling. To compare these three methods, we focus on five aspects: use of precedent information, coordination complexity, simulated product designs, flexibility, and throughput time. For the throughput time aspect we derive mathematical formulas and give relations between the throughput times of the three coordination methods. At the end of this part we summarize the results and give recommendations on the choice of a suitable coordination method.

The third part of this paper contains a case-study, in which the CMM approach is applied to the design process of a color picture tube.

Key Words: design optimization, coordination, simulation, collaboration, black box, metamodel.

1. Introduction

Simulation tools, like FEM or CFD, are frequently used nowadays in the design process to predict product or process characteristics. Because of the complexity of many of those simulation tools there are often no explicit input–output formulas known. This is why these tools are referred to as black boxes. Further, simulation runs are often very time-consuming, so the number of simulated scenarios is limited in practice.

Several sequential optimization methods have been introduced in the literature to deal with optimization involving expensive simulations. These methods try to find an optimal product design by means of derivative-free optimization and search methods; see, e.g., [6,7,10,20,26].

Alternatively, it has been proposed to replace the black boxes by approximation models, also called metamodels; see, e.g., [2,5,11,12]. Equivalent terms that appear in the literature are compact models, surrogate models, and response surface models. With such metamodels we can evaluate product designs relatively fast and thus gain insight in the product and find optimal and robust product designs in an admissible time.

The manufacturing industry produces a lot of complex products that consist of several coupled components. Due to the complexity of many components, their design is often distributed over specialized design teams. Each of those teams uses its own black box(es) to evaluate the component designs. Because of the coupling among the components, their black boxes are also coupled. Up to now, mainly sequential optimization methods have been proposed in the literature to deal with these kind of problems. In engineering practice, Multidisciplinary Design Optimization (MDO) techniques are often used to deal with coupled design problems, e.g., see [8,14,18,21]. These techniques are based on optimization procedures that iteratively solve several small optimization problems in order to gradually converge to the optimal solution.

In practice, however, we are not just interested in the optimal solution, but we also want to gain insight in the product behavior in order to design a reliable product.

*Author to whom correspondence should be addressed.
E-mail: b.g.m.husslage@uvt.nl

1063-293X/03/04 0267–12 $10.00/0 DOI: 10.1177/1063293X03039895
Downloaded from http://cer.sagepub.com at Universiteit van Tilburg on May 14, 2007
© 2003 SAGE Publications
© 2003 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

This is accomplished by an efficient construction of a metamodel for the product as a whole. Because of the large number of design (or input) and response (or output) parameters of the product, we cannot do this all-at-once, e.g., using the metamodeling approach. Instead, we exploit the product's architecture by constructing metamodels for all black boxes and combine these models into a metamodel for the product. However, due to the coupling among black boxes, we must carefully coordinate this modeling process. We do this by defining a coordination method that controls the order of the simulation runs and the construction of metamodels. This coordination method enables us to construct metamodels for all black boxes in an efficient way. The set of these models then implicitly forms the required metamodel for the product. We call this approach Collaborative Metamodeling (CMM).

The paper is organized as follows. Section 2 gives a general framework for CMM. Here the steps to be followed and the questions and problems that arise at these steps are discussed. Section 3 focuses on the first step of the CMM approach. It introduces the notion of coordination methods and defines three different methods. Section 4 discusses five important aspects of coordination methods and uses these aspects to compare the three methods. Section 5 contains a case study in which the CMM approach is applied to the design process of a color picture tube. Finally, Section 6 gives the conclusions and some topics for further research. The Appendix contains the mathematical derivations for the throughput time aspect in Section 4.5.

2. Collaborative Metamodeling

In case of a single black box the design optimization process can be divided into four basic steps; Problem Specification, Design and Analysis of Computer Experiments (DACE), Metamodeling, and Design Optimization [22]. Our CMM approach uses the same steps, be it that some extra work has to be performed at each step, in order to deal with the relations among black boxes.

Next, the four steps in our CMM approach are discussed, as well as the problems that are encountered when applying this procedure to complex product design problems.

2.1 Step 1: Problem Specification

At this step we examine the most important features of the product design problem. We investigate the product's architecture, i.e., we determine all components the product consists of and the relations between them. This includes the definition of all design and response parameters of the black boxes, as well as the determination of the simulation times and the number of simulated component designs needed or possible. The latter holds in cases where there is a budgetary maximum on the time spent on simulation. Then it may be wise to cluster several black boxes into larger black boxes, in order to reduce the total simulation time. A methodology for clustering, in order to reduce the design project makespan, can be found in [16]. Note, however, that clustering can have unwanted effects on the construction of metamodels, which may not weigh up against the reduction in simulation time.

Further, we investigate the restrictions on design parameter settings or combinations of design parameters. All these restrictions together form the design space for the product. Attention has to be paid to design parameters that are input to multiple black boxes, due to the fact that this allows restrictions on one black box to affect other black boxes, and, hence, their design spaces.

Another issue is that response parameters of a black box may be used as input to another black box. Depending on the coordination method, values of response parameters may not be known before the simulations take place. Then we can only use the expected parameters' lower and upper bounds that are provided by the design teams.

The problems mentioned above ask for a proper coordination of the product design process. In Section 3.2 we define coordination methods to deal with this.

2.2 Step 2: Design and Analysis of Computer Experiments (DACE)

Once all parameter ranges are known (or estimated), we have information about the design spaces of the components and the product. Next, we have to decide which component designs to simulate. Every design team will have to simulate several designs for their component. We will call such a set of simulation points a simulation scheme.

The construction of a simulation scheme depends on the chosen coordination method, since this method determines whether there is information available prior to the simulation process of a particular component. However, there are some basic requirements that we would like our simulation scheme to meet. For one, we like our simulation scheme to be space-filling, i.e., the simulation points to be distributed over the design space, in order to gain information about the whole design space. Further, we like to have a noncollapsing simulation scheme. This means that every simulation point should have different settings for the design parameters. When, after the simulation process, it turns out that a particular design parameter is not important, no simulation results have collapsed, and we do not loose valuable simulation time. For the construction of space-filling, noncollapsing simulation schemes, see [17]

Downloaded from http://cer.sagepub.com at Universiteit van Tilburg on May 14, 2007
© 2003 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

in cases where the design space is a box region and [22] for nonbox regions.

Depending on the coordination method we may be able to carry out system level simulations, i.e., the simulation of designs of the final product. Section 4.3 gives reasons why we want to do this. The construction of the corresponding simulation schemes is subject of current research.

2.3 Step 3: Metamodeling

When all points in a simulation scheme have been simulated, we can use the observed simulation results to construct a metamodel. The model types most often used are polynomials, response surface approximations, neural networks, and Kriging models, e.g., see [2].

After constructing metamodels for all black boxes, we can use the relations between the components to combine these models into a metamodel for the product. Combining the approximation models for the black boxes can lead to serious error propagation, which can result in a poor metamodel for the product. Therefore, we should validate the product's metamodel, e.g., using cross-validation, see [13,22]. Should the metamodel appear to be invalid, then we need to carry out an additional set of simulation runs to improve this model. The problem then is to determine which components to grant extra simulation runs and which designs to simulate. This is subject of further research.

2.4 Step 4: Design Optimization

The last step in our CMM approach is to use the product's metamodel and optimization techniques to gain insight in the product and find an optimal product design that fulfills all restrictions, i.e., lies within the product's design space. Due to the fact that the metamodel is an explicit function, function evaluations are relatively fast, and, hence, approaches like Mathematical Programming and Global Optimization can be applied to the problem; see, e.g., [4,19].

Since the resulting optimal product design will be an approximation of the real (unknown) optimum, it is wise to simulate the found product design. When the observed responses do not deviate too much from the responses estimated by the metamodel, we can be confident in having found a good product design.

During manufacture of the product design, the design parameters may be subject to noise, e.g., due to small errors in the construction of components, resulting in a somewhat different design. To deal with this last problem, as well as possible modeling errors in the metamodels, it is wise to take robustness into account. See [23] for a more complete discussion and application of robust design.

With this last step we finish the first part of this paper. We now turn to the problem specification and to coordination methods in particular.

3. Coordination Methods

This section focuses on the first step of the CMM approach, i.e., the problem specification. The problem's architecture is discussed and three coordination methods are defined that enable an efficient construction of metamodels for the coupled black boxes.

3.1 Product's Architecture

Every black box within the product's architecture has several design and response parameters. The design parameters can be divided into local design parameters and linking design parameters. Local design parameters are input to a single black box, whereas linking design parameters are input to multiple black boxes. There is also a special type of input parameter, i.e., the response input parameter, which is a black box response that is input to other black boxes. Figure 1 shows the design and response parameters, as well as all possible couplings, for two coupled black boxes.

From Figure 1 it can be seen that black boxes can be coupled by linking design parameters and response input parameters. The presence of response input parameters gives rise to the need for a coordination method; see Section 3.2.

We can represent this type of coupling by a directed graph in which the nodes represent the black boxes and the arcs represent their relations or couplings. We assume that there exist no cycles in the directed graph. This is a common assumption in the literature, e.g., see [1,25]. More important, this assumption is substantiated by problems found in practice. The nonexistence of cycles gives our directed graph a forward structure, i.e., there is an explicit precedence ordering of the black boxes. Therefore, we refer to the directed graph as a black box chain. See Figure 2 for an example of such a chain.

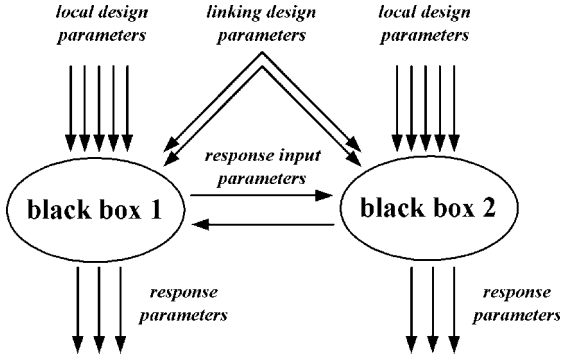

Figure 1. Design and response parameters for two coupled black boxes.

Downloaded from http://cer.sagepub.com at Universiteit van Tilburg on May 14, 2007
© 2003 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

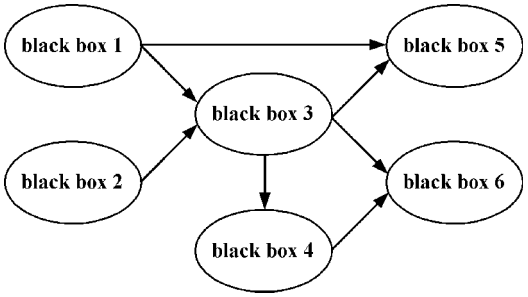

Figure 2. Black box chain with six black boxes coupled by response input parameters.

In Figure 2 an arc represents one or more response input parameters. Note that there may be multiple independent black box chains within the product's architecture. All these chains can be dealt with concurrently in the way described here.

3.2 Three Coordination Methods

As mentioned in Section 2.1, the coupling among black boxes asks for some kind of coordination of the product design process. We define a coordination method to be a rule that determines the order in which simulation runs are carried out and metamodels are constructed by the different component design teams. In this paper we introduce and analyze the following three coordination methods.

Parallel simulation Every black box is dealt with separately, i.e., independent of all others. Linking and response input parameters are seen as local design parameters and a simulation scheme is constructed based on local restrictions only. Every design team carries out their simulations concurrently.

Sequential simulation In Sequential simulation we use simulation results from black boxes preceding the one in question. Once a component design has been simulated, the simulation results are transferred to all its successors (if any). When a particular black box has received the simulation results from all its predecessors, a simulation run is carried out, i.e., one component design is simulated. This procedure is repeated until the needed number of simulations is reached.

It is important to note that the simulations at every design team are carried out *one-by-one*, following the precedence ordering in the black box chain and using simulation results of predecessors.

Sequential modeling This method closely resembles Sequential simulation. The main difference is that the simulation runs for a black box are carried out *all-at-once* and the simulation results, along with the constructed metamodel, are transferred to all its successors.

Again, the precedence ordering in the black box chain is followed, but now design teams have to wait until their predecessors are completely finished with their simulation and metamodeling processes. However, note that this provides them with much information about their predecessors.

In the next section we look at the coordination methods more closely. We define and analyze several aspects, in order to compare the three coordination methods.

4. Aspects of Coordination Methods

Once all points in a simulation scheme for a black box have been simulated, we can construct a metamodel for that black box, based on the simulation results found. These metamodels are then used in the product design optimization process. However, the construction and validation of metamodels depends heavily on the availability of proper data. This, in order, depends on the way the simulations have been carried out, and, even more important, which simulations have been carried out. Clearly, the chosen coordination method plays a major role in this.

In this section we compare the three coordination methods. As a measure for comparison we look at the following five aspects:

1. Use of precedent information;
2. Coordination complexity;
3. Simulated product designs;
4. Flexibility;
5. Throughput time.

For every aspect, after defining it, we discuss the effect of the three coordination methods on this aspect and compare the methods. Finally, in Section 4.6, we summarize the results found and give recommendations on the choice of a coordination method.

4.1 Use of Precedent Information

This aspect refers to the use of simulation and modeling results from preceding simulation tools. The latter results can help in the determination of designs that are expected to yield the most valuable information about the components. Note that in case of response input parameters the use of precedent information is a necessity to obtain simulated product designs; see Section 4.3. Clearly, both sequential coordination methods use precedent information by means of the response input parameters. In Parallel simulation there is no precedent information used.

4.2 Coordination Complexity

Coordination complexity refers to the amount of communication and time that is needed to implement

Downloaded from http://cer.sagepub.com at Universiteit van Tilburg on May 14, 2007
© 2003 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

a coordination method. It also includes extra costs that are incurred by, e.g., the need for an automated communication system.

In Parallel simulation every design team operates independently and there is no need for a complex organizational structure; see, e.g., [15], where managing the simultaneous execution of two coupled development phases plays a central role.

In Sequential simulation, communication is needed after each global design simulation, see Appendix A.2, at every black box. Therefore, this coordination method results in a complex coordination process that needs sophisticated communication methods, which have to be supported by the design tools. Communication between design teams is also required in Sequential modeling, be it only after a complete simulation scheme has been run. Hence, the coordination process is relatively simple.

4.3 Simulated Product Designs

A product design is a particular setting of all the design parameters in the product specification. When we simulate such a setting, we obtain a simulated product design or system level simulation. However, the simulations are carried out by simulation tools at the component level, which all have a subset of the product's design parameters as input. Further, linking design parameters and response input parameters create overlap in the sets of design parameters and couple the simulation tools. In order to obtain a simulated product design, we must use the same setting for the linking design parameters at every simulation tool and use the simulation results of response input parameters as settings in succeeding simulation tools.

Clearly, it is not possible to obtain simulated product designs in Parallel simulation, since all linking design and response input parameters are seen as local design parameters. Simulated product designs can be obtained in Sequential simulation and Sequential modeling, since these coordination methods use the simulation results from preceding simulation tools as inputs, via the response input parameters. Besides this, it is also necessary to use the same settings for the linking design parameters at all simulation tools. This can be accomplished by the construction of coupled simulation schemes, which is subject of current research.

Obtaining simulated product designs may require some effort, but it is a great help in the product design process. We get information about the characteristics of feasible product designs, which gains insight in the product, and can be used in the product design optimization, e.g., by using the characteristics as lower bounds. Further, simulating product designs also increases the credibility of the used optimization and robust design approaches. Finally, it shows the design teams the effects of their components on the final product, and it is more easy to compare component designs between two teams of designers.

4.4 Flexibility

Flexibility of a coordination method means that it does not take a lot of effort to validate or adjust the constructed metamodels, when a small change is made to one or more simulation tools.

Metamodels are based on results found by the simulation tools. Hence, should a simulation tool be adjusted, e.g., due to changes in the underlying component, then the constructed metamodel will probably no longer be valid. In Parallel simulation, this can be fixed by simulating an extra set of designs for the corresponding component, since the metamodels for the various black boxes are constructed independently.

However, at the two sequential coordination methods, one must be more careful, since coupling among simulation tools is preserved in the construction of metamodels. Therefore, invalidity of one metamodel can affect the validity of the metamodels of all its successors. Since small changes can require much effort in the validation and, possibly, adjustment of many metamodels, the sequential coordination methods are not flexible with respect to changes in the simulation tools, while Parallel simulation is flexible.

4.5 Throughput Time

The throughput time of a coordination method is defined as the total time it takes to carry out all simulation runs needed to construct metamodels for every black box in the chain. From a time-to-market perspective it is desirable to have short product development times, so the throughput time should preferably be small. Since the construction time of metamodels is assumed to be negligible, relative to the simulation run time, we ignore it in this analysis.

In the Appendix we derive formulas for the throughput times of all three coordination methods and give the relations between them. We show that Parallel simulation always leads to the shortest throughput time, Sequential simulation takes a longer time, and Sequential modeling the longest. Using the formulas given in the Appendix one can simply compute the exact throughput time for each coordination method at a particular problem instance. This information, along with the four other aspects, can be used to make a decision about the coordination method to use.

4.6 Summary of Aspects

We now summarize the results found above. Table 1 gives the five aspects discussed in Sections 4.1–4.5, as well as the following scores for each of the three

Downloaded from http://cer.sagepub.com at Universiteit van Tilburg on May 14, 2007
© 2003 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

Table 1. Comparison of the three coordination methods with respect to five aspects.

Aspect/ Coordination Method	Parallel Simulation	Sequential Simulation	Sequential Modeling
Use of precedent information	− −	+	++
Coordination complexity	++	− −	+
Simulated product designs	−	++	++
Flexibility	++	− −	− −
Through out time	++	+	− −

coordination methods at these aspects. Two pluses (+ +) indicate that the coordination method has a positive effect on a particular aspect; one plus (+) indicates a moderately positive effect. With one minus (−) the effect of the coordination method on a particular aspect is slightly negative; with two minuses (−−) this effect is negative. Note that in Table 1 a (moderately) positive effect (+ or + +) on the coordination complexity means that the coordination process is *not* complex.

The main advantages of Parallel simulation are a small throughput time, much flexibility, and a lack of complexity at the coordination process. However, for the designers it is important that they obtain accurate metamodels for the components and the product. Parallel simulation may need several extra simulation runs, besides the designs already simulated, to include the coupling among black boxes properly in the metamodels. Further, there should be simulated product designs, for reasons mentioned in Section 4.3. This requirement makes the sequential coordination methods more suitable than the parallel coordination method, at least in those cases with response input parameters.

Choosing between the two sequential coordination methods mainly depends on the throughput time and the availability of good means of communication among the design teams. Using Sequential simulation results in a more complex coordination process, whereas Sequential modeling yields a larger throughput time. Therefore, when dealing with large simulation times and an automated communication system, the Sequential simulation method is preferable. Sequential modeling is a good choice when communication among design teams is hard and the simulation times are not too large.

Of course, the determination of the best coordination method is not so strict and depends on the kind of product design problem we are dealing with. This is why a careful study of all aspects for each of the three coordination methods is extremely important. This can be done by using Table 1, along with the discussion in Sections 4.1–4.5, and the formulas derived in the Appendix.

Section 2.3 mentions the problem of constructing proper metamodels. Since the initial sets of simulation runs may not suffice to construct proper metamodels for each black box, a two-stage simulation procedure is often used. For the first stage we advise to use Parallel simulation and to run all simulation schemes concurrently. This gives a good idea about the black box behavior and the most important parts of the component and product design spaces. In the second stage we can combine the simulation results with a sequential method, and simulate extra sets of component designs. These extra simulation runs will give more insight into the most important parts of the product design space. Therefore, the resulting metamodels will give good representations for the whole product design space, but they still emphasize the most important parts of it.

5. Case Study: Color Picture Tube Design

This section summarizes the results found at a successful application of our Collaborative Metamodeling approach to the design process of a color picture tube at LG. Philips Displays in Eindhoven. For a detailed discussion of this application the reader is referred to [24].

Contents of this study was the collaborative design of several aspects of the shadow mask and screen for a color picture tube. The problem specification resulted in the problem structure that is given in Figure 3.

In this figure there are four black boxes, which are represented by the ellipses. The numbers in brackets give the time needed per simulation run and the numbers above the black boxes give the number of simulation runs performed. Rectangles represent design parameters, with their corresponding numbers given in italic. Note that MicMac center and MicMac northeast have local design parameters only, whereas Landing also has linking design parameters. Microphony has linking, as well as response input parameters. Table 2 summarizes these data.

Parallel simulation was chosen as coordination method. The main reason for this choice was that the design optimization tool COMPACT [22] could be used to find simulation schemes and to construct metamodels for all four black boxes. The quality of the constructed metamodels, given by the cross-validation RSME, as well as the number of simulations performed, can be found in the last two columns of Table 2.

Combining these individual metamodels leads to the construction of a system level metamodel. This was done using COMPACT-CO, the collaborative version of COMPACT, see [24]. For the validation of the system level metamodel a test set was used, that was taken from the predicted feasible product design space.

Downloaded from http://cer.sagepub.com at Universiteit van Tilburg on May 14, 2007
© 2003 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

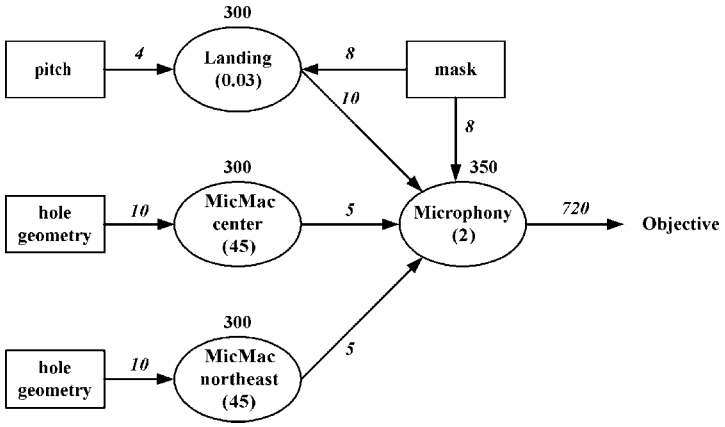

Figure 3. The problem structure.

Table 2. Black box characteristics and metamodel validation results.

Black Box	# Design Parameters	# Response Parameters	# Simulations	Average Relative CV-RMSE (%)
Landing	12	10	300	2.86
MicMac center	10	5	300	3.06
MicMac northeast	10	5	300	4.08
Microphony	28	720	350	8.60

Since only quadratic metamodels were used, we got a quadratic optimization problem. CONOPT [9] and a multistart technique were used to solve for a global optimum. The optimal Microphony design found turned out to be an improvement of 50% with respect to the current design. In order to test the robustness of the design, Monte-Carlo analysis was applied to the design. Since metamodels are explicit functions, this type of analysis is very fast.

This study showed that Collaborative Metamodeling improves the insight in the design problem. Further, the constructed metamodels can be used for Monte-Carlo analysis, to ensure that the product design remains valid under small perturbations.

However, in this study it can be expected that the quality of the metamodel for Microphony is improved when one of the two sequential coordination methods is used, since the latter black box has response input parameters coming from the three other black boxes. Since Microphony directly affects the objective this would probably lead to a better system level metamodel, given that the same number of simulations is used. However, using one of the sequential coordination methods makes the coordination process much more complex; the question is whether these costs weigh up against the possibility of a better metamodel.

6. Conclusions and Further Research

In order to gain insight in the behavior of a product and its components, we replace time-consuming simulation tools by metamodels. The construction of such models is based on results found by simulating component designs. Products that consist of several components usually have several of these time-consuming simulation tools, or black boxes, that are used in the component design processes. Coupling among components requires coupling among the black boxes, and results in a need for coordinating the simulation, metamodeling, and optimization process.

This paper introduced a framework, called Collaborative Metamodeling (CMM), to deal with product optimization in case of coupled, time-consuming simulation tools. The four steps of the CMM approach: Problem Specification, Design and Analysis of Computer Experiments (DACE), Metamodeling, and Design Optimization, were discussed, as well as problems that may arise at each of these steps.

We focused on the first step of the CMM approach, and the coordination of the simulation and metamodeling process in particular. Three coordination methods were introduced: Parallel simulation, Sequential simulation, and Sequential modeling. The five aspects: use of precedent information, coordination complexity, simulated product designs, flexibility, and throughput time, were used to compare the three methods. The results of this comparison can be found in Table 1.

This table, along with the throughput time formulas that were derived in the Appendix, can be used to determine the best coordination method for a specific product design problem. Next, the method chosen can be used to construct simulation schemes for all coupled simulation tools. The latter construction method is subject of current research. Based on the simulation results, metamodels can be constructed for all black boxes. These metamodels can then be used for finding an optimal or robust product design.

Downloaded from http://cer.sagepub.com at Universiteit van Tilburg on May 14, 2007
© 2003 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

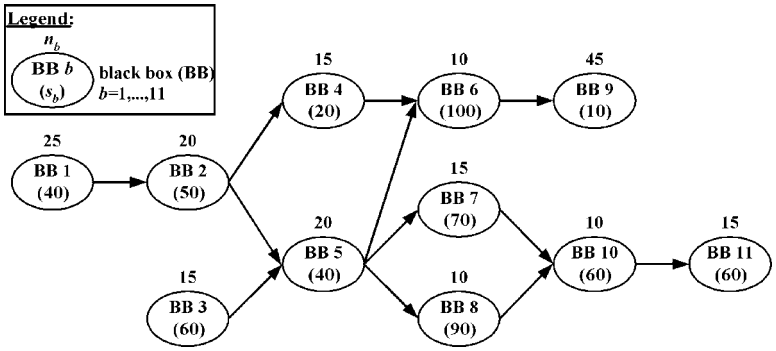

Figure 4. Numerical example of coupling among eleven black boxes by response input parameters.

To end the paper we gave a successful application of our CMM approach to the design process of a color picture tube at LG. Philips Displays in Eindhoven.

Acknowledgments

The authors wish to thank Jack Kleijnen and the anonymous referees for their valuable comments on a previous version of this paper.

Appendix

This appendix derives mathematical formulas for the throughput time aspect in Section 4.5. Further, we clarify the throughput time computation for all three coordination methods, by means of a numerical example. Since the construction time of metamodels is assumed to be negligible, relative to the simulation run time, we ignore it.

Throughout this appendix we use the following notation to indicate the black boxes and their characteristics:

B : set of black boxes, $B = \{1, 2, \ldots\}$;
P_b : set of all black boxes that directly precede black box $b \in B$;
B_e : set of all black boxes with no successors, i.e., $B_e = B \backslash (\cup_{b \in B} P_b)$;
n_b : number of required simulation runs at black box $b \in B$;
s_b : time per simulation run at black box $b \in B$.

Using this notation we can always number the black boxes in such a way that their numbering reflects the precedence ordering in the chain, i.e., $b \notin P_{\tilde{b}}$ if $b \geq \tilde{b}$. Note that $P_b = \emptyset$ if black box $b \in B$ is at the beginning of the chain; $b \in B_e$ if b is at the end of the chain.

Figure 4 gives the numerical example that is used to clarify the throughput time computation. The figure shows eleven black boxes that are coupled by response input parameters. The actual time unit of the simulation times is not important for our discussion; we let it be minutes.

A.1 Parallel Simulation

With Parallel simulation all design teams carry out their simulations concurrently. Hence, the corresponding throughput time, denoted by TT_{parallel}, is equal to the maximum of the total simulation times at every black box:

$$TT_{\text{parallel}} = \max_{b \in B} n_b s_b. \tag{1}$$

We call a black box a bottleneck when a small increase of its simulation time s_b results in an increase of the throughput time. In Parallel simulation the bottlenecks are all black boxes $\hat{b} \in B$ that satisfy

$$\hat{b} = \arg\max_{b \in B} n_b s_b. \tag{2}$$

For the black box chain in Figure 4 it can readily be computed that TT_{parallel} is equal to 1050 min and that black box 7 (BB 7) forms the bottleneck.

A.2 Sequential Simulation

In Sequential simulation the results of simulating a component design at a particular black box are passed down to its succeeding black boxes. This process can be viewed as a flow of information objects through the entire black box chain, where an information object carries the simulation results from preceding black boxes. Therefore, the simulations that make use of this information are referred to as global (component) design simulations. The maximum number of this type of simulation runs, say $\hat{n}$, is restricted by the minimum number of required simulated designs per black box, i.e., $\hat{n} = \min_{b \in B} n_b$. Because $\hat{n}$ is a minimum, there may be several black boxes that require more simulations. The simulation results of the latter are used only locally, i.e., at a certain black box, and therefore these simulations

Downloaded from http://cer.sagepub.com at Universiteit van Tilburg on May 14, 2007
© 2003 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

are referred to as local design simulations. Clearly, every black box $b \in B$ invokes $\hat{n}$ global and $n_b - \hat{n}$ local design simulations.

Focusing on the global design simulations, we see that such a simulation run can only start when all preceding black boxes have finished their global design simulation run. In order to compute the throughput time of the Sequential simulation method we first introduce the throughput time function $f_b(n)$. This function gives the minimal time it takes for n global design simulations to be finished at black box $b \in B$. We can then write

$$f_b(n) = \max\left\{ s_b + \max_{\tilde{b} \in P_b} f_{\tilde{b}}(n), \quad s_b + f_b(n-1) \right\}, \quad n \geq 1. \tag{3}$$

The interpretation of this formula is that black box b has to wait until all its predecessors have finished n global design simulations and itself has simulated $n-1$ global designs, before it can start with the nth simulation run. Further, note that $f_b(0) = 0$ implies

$$f_b(1) = s_b + \max_{\tilde{b} \in P_b} f_{\tilde{b}}(1). \tag{4}$$

This equation computes the longest path up to black box b, e.g., see [3], starting at a black box in the beginning of the chain.

Because Equation (3) is dynamic in the variables b and n, we can rewrite it:

$$\begin{aligned} f_b(n) &= \max\left\{ s_b + \max_{\tilde{b} \in P_b} f_{\tilde{b}}(n), \right. \\ &\quad \max\left\{ 2s_b + \max_{\tilde{b} \in P_b} f_{\tilde{b}}(n-1), \right. \\ &\quad \left. \left. 2s_b + f_b(n-2) \right\} \right\} \\ &= \max\left\{ s_b + \max_{\tilde{b} \in P_b} f_{\tilde{b}}(n), \quad 2s_b + \max_{\tilde{b} \in P_b} f_{\tilde{b}}(n-1), \right. \\ &\qquad \left. 2s_b + f_b(n-2) \right\} \\ &= \max\left\{ s_b + \max_{\tilde{b} \in P_b} f_{\tilde{b}}(n), . \quad 2s_b + \max_{\tilde{b} \in P_b} f_{\tilde{b}}(n-1), \right. \\ &\quad 3s_b + \max_{\tilde{b} \in P_b} f_{\tilde{b}}(n-2), \ldots, \\ &\qquad \left. (n-1)s_b + \max_{\tilde{b} \in P_b} f_{\tilde{b}}(2), \quad ns_b + \max_{\tilde{b} \in P_b} f_{\tilde{b}}(1) \right\} \\ &= \max_{k=1,\ldots,n} \left\{ ks_b + \max_{\tilde{b} \in P_b} f_{\tilde{b}}(n+1-k) \right\}, \quad n \geq 1. \end{aligned} \tag{5}$$

It can be proven that $f_b(n)$ is convex in $n \geq 1$. Therefore, Equation (5) can be simplified to the following maximum function, that is only dynamic in the variable b.

$$\begin{aligned} f_b(n) &= \max_{k \in \{1,n\}} \left\{ ks_b + \max_{\tilde{b} \in P_b} f_{\tilde{b}}(n+1-k) \right\} \\ &= \max\left\{ s_b + \max_{\tilde{b} \in P_b} f_{\tilde{b}}(n), \quad ns_b + \max_{\tilde{b} \in P_b} f_{\tilde{b}}(1) \right\} \\ &\overset{\text{Eq.(4)}}{=} \max\left\{ s_b + \max_{\tilde{b} \in P_b} f_{\tilde{b}}(n), \quad (n-1)s_b + f_b(1) \right\}, \\ &\qquad n \geq 2. \end{aligned} \tag{6}$$

Next, we define the two sets

$C_{b,n}$: set of possible bottlenecks up to black box $b \in B$, when simulating n global designs;

C_n : set of bottlenecks when simulating n global designs.

Note that the black box chain structure can cause the sets $C_{b,n}$ and C_n to differ significantly for distinct values of n. For $n = 1$ we use Equation (4) and $C_{b,1} = \{b\} \cup \cup_{\tilde{b} \in I} C_{\tilde{b},1}$, with $I = \{\tilde{b} | \tilde{b} \in P_b; f_{\tilde{b}}(1) = f_b(1) - s_b\}$, to compute $f_b(1)$ and the corresponding bottlenecks for all black boxes $b \in B$. Combining this information with Equation (6) enables us to compute the throughput time $f(n)$ and the corresponding sets $C_{b,n}$ and C_n for every arbitrary integer $n \geq 2$ as follows.

for $b \in B$ **do**

$$ptf_b(n) = \max\left\{ s_b + \max_{\tilde{b} \in P_b} f_{\tilde{b}}(n), \quad (n-1)s_b + f_b(1) \right\}; \tag{7}$$

$$C_{b,n} = \begin{cases} C_{b,1} \cup \bigcup\limits_{\tilde{b} \in I} C_{\tilde{b},n} & \text{if } f_b(n) = (n-1)s_b + f_b(1) \\ \{b\} \cup \bigcup\limits_{\tilde{b} \in I} C_{\tilde{b},n} & \text{otherwise} \end{cases}; \tag{8}$$

where $I = \{\tilde{b} | \tilde{b} \in P_b; f_{\tilde{b}}(n) = f_b(n) - s_b\}$;
end

$$f(n) = \max_{b \in B_e} f_b(n); \tag{9}$$

$$C_n = \bigcup_{b \in J} C_{b,n}; \text{ where } J = \{b | b \in B_e; f_b(n) = f(n)\}. \tag{10}$$

Using Equations (7)–(10), we can compute the minimal time that is needed to simulate all $\hat{n}$ global designs, i.e., $f(\hat{n})$. To compute the throughput time, we must include the time needed to simulate the local

Downloaded from http://cer.sagepub.com at Universiteit van Tilburg on May 14, 2007
© 2003 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

designs at every black box. In this respect, note that a black box may be idle, i.e., not simulating, for several periods of time during the whole simulation process of global designs. When it is possible to stop a simulation run at some point in time and later on proceed from that point, we call the simulation run preemptive. In this case, we can start simulating local designs within the idle periods.

The throughput time of the Sequential simulation method, denoted by $TT^{\text{pre}}_{\text{seqsim}}$, is then given by

$$TT^{\text{pre}}_{\text{seqsim}} = \max\left\{ \max_{b\in B} n_{\text{b}} s_{\text{b}}, \quad f(\hat{n}) \right\}$$

$$\overset{\text{Eq.(1)}}{=} \max\left\{ TT_{\text{parallel}}, \quad f(\hat{n}) \right\}. \tag{11}$$

In case $TT^{\text{pre}}_{\text{seqsim}} = TT_{\text{parallel}}$, the black boxes $\hat{b} \in B$ that satisfy Equation (2) form bottlenecks. If $TT^{\text{pre}}_{\text{seqsim}} = f(\hat{n})$, the black boxes $b \in C_{\hat{n}}$ are bottlenecks. Note that in the latter case not all bottlenecks may have the same impact on the throughput time. However, the exact impact of every bottleneck can easily be computed.

Unfortunately, simulation runs are often nonpreemptive. Switching between different component designs within the simulation process may also cost much time, and may, therefore, not be very practical. Further, waiting until all global design simulations are finished gives a design team much information to determine which local designs to simulate. For these reasons we suggest to use nonpreemptive simulation runs and suggest to simulate all local designs after the global design simulations are finished. The throughput time then becomes

$$TT_{\text{seqsim}} = \max_{b\in B}\left\{ f_{\text{b}}(\hat{n}) + (n_{\text{b}} - \hat{n}) s_{\text{b}} \right\}. \tag{12}$$

In this case the bottlenecks are given by the set $C^*_{\hat{n}} = \cup_{\text{b}\in I} C_{\text{b},\hat{n}}$, with $I = \left\{ b | f_{\text{b}}(\hat{n}) + (n_{\text{b}} - \hat{n}) s_{\text{b}} = TT_{\text{seqsim}} \right\}$. As above, the impact of these bottlenecks may vary.

For the numerical example in Figure 4 we can use Equations (7)–(10) to compute that

$$f(n) = \begin{cases} 250 + 90n & \text{if } n \le 11 \\ 140 + 100n & \text{if } n \ge 11. \end{cases} \tag{13}$$

From Equation (13) and the fact that $\hat{n} = 10$ we get $f(\hat{n}) = 1150$. Recall that $TT_{\text{parallel}} = 1050\,\text{min}$, so Equation (11) results in $TT^{\text{pre}}_{\text{seqsim}} = 1150\,\text{min}$. Further, because $f_{11}(\hat{n}) = f(\hat{n})$ it follows from Equation (10) that the bottlenecks are given by the set $C_{\hat{n}} = C_{11,\hat{n}} = \{1, 2, 5, 8, 10, 11\}$. Using Equation (12) we find that TT_{seqsim} is equal to 1490 min. Because $f_9(\hat{n}) + (n_9 - \hat{n}) s_9 = TT_{\text{seqsim}}$, all black boxes in the set $C^*_{\hat{n}} = C_{9,n} = \{1, 2, 5, 6, 9\}$ are bottlenecks.

A.3 Sequential Modeling

In Sequential modeling, the required simulations are carried out all-at-once, before results are passed down to succeeding black boxes. Therefore, the throughput time, denoted by TT_{seqmod}, is equal to the longest path in the black box chain, when we take the total simulation time per black box, i.e., $n_{\text{b}} s_{\text{b}}$, on the nodes of the directed graph:

$$TT_{\text{seqmod}} = \max_{\hat{B}\subset B} \sum_{b\in\hat{B}} n_{\text{b}} s_{\text{b}}, \tag{14}$$

where $\hat{B}$ is a path in the chain.

In this equation a path is defined as a sequence of black boxes starting at a beginning of the chain, so at a black box b for which $P_{\text{b}} = \emptyset$, and ending at an end of the chain, so at a black box $b \in B_{\text{e}}$. All black boxes on a longest, or critical, path form bottlenecks. In Figure 4, with $n_{\text{b}} s_{\text{b}}$ on the nodes, the black boxes 1, 2, 5, 7, 10 and 11 form the (unique) longest path, and, therefore, the bottlenecks. The corresponding throughput time is equal to $TT_{\text{seqmod}} = 5350\,\text{min}$.

A.4 Throughput Time Relations

From the above observations we can derive general relations among the throughput times of the different coordination methods:

$$TT_{\text{parallel}} \overset{\text{Eq.(11)}}{\le} TT^{\text{pre}}_{\text{seqsim}} \overset{\text{Eq.(12)}}{\le} TT_{\text{seqsim}} \le TT_{\text{seqmod}}. \tag{15}$$

The first two inequalities readily follow from Equations (11) and (12). It can easily be proven that the last inequality also holds.

References

1. Assine, A., Falkenburg, D. and Chelst, K. (1999). Engineering Design Management: An Information Structure Approach, *International Journal of Product Research*, **37**: 2957–2975.
2. Barton, R.R. (1998). Simulation Metamodels, In: *Proceedings of the 1998 Winter Simulation Conference*, pp. 167–176, Washington, USA.
3. Bazaara, M.S., Jarvis, J.J. and Sherali, H.D. (1990). *Linear Programming and Network Flows*, John Wiley & Sons, New York.
4. Birge, J.R. and Murty, K.G. (eds), (1994). *Mathematical Programming: State of the Art*, University of Michigan Press, Ann Arbor, MI.
5. Booker, A.J., Dennis, J.E., Frank, P.D., Serafini, D.B., Torczon, V. and Trosset, M.W. (1999). A Rigorous Framework for Optimization of Expensive Functions by Surrogates, *Structural Optimization*, **17**: 1–13.

Downloaded from http://cer.sagepub.com at Universiteit van Tilburg on May 14, 2007
© 2003 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

6. Brekelmans, R., Driessen, L., Hamers, H. and den Hertog, D. (2001). A New Sequential Optimization Approach to Product and Process Design Involving Expensive Simulations, In: *Proceedings of the Third ASMO UK/ISSMO Conference on Engineering Design Optimization: Product and Process Improvement*, pp. 49–52.
7. Conn, A.R., Scheinberg, K. and Toint, P.L. (1997). Recent Progress in Unconstrained Nonlinear Optimization without Derivatives, *Mathematical Programming*, **79**(1–3): 397–414.
8. Cramer, E.J., Dennis Jr., J.E., Frank, P.D., Lewis, R.M. and Shubin, G.R. (1994). Problem Formulation for Multidisciplinary Optimization, *SIAM Journal on Optimization*, **4**: 754–776.
9. Drud, A.S. (1994). CONOPT – A Large Scale GRG Code. *ORSA Journal of Computing*, **6**: 207–216.
10. Glover, F., Kelly, J.P. and Laguna, M. (1996). New Advances and Applications of Combining Simulation and Optimization, In: *Proceedings of the 1996 Winter Simulation Conference*, pp. 144–152.
11. Jones, D., Schonlau, M. and Welch, W. (1998). Efficient Global Optimization of Expensive Black Box Functions, *Journal of Global Optimization*, **13**: 455–492.
12. Kleijnen, J.P.C. (1987). *Statistical Tools for Simulation Practitioners*, Marcel Dekker, Inc., New York.
13. Kleijnen, J.P.C. and Sargent, R.G. (2000). A Methodology for Fitting and Validating Metamodels in Simulation, *European Journal of Operational Research*, **120**: 14–29.
14. Kodiyalam, S. and Sobieszczanski-Sobieski, J. (2001). Multidisciplinary Design Optimization – Some Formal Methods, Framework Requirements, and Application to Vehicle Design, *International Journal of Vehicle Design*, Special Issue: 3–22.
15. Krishman, V. (1996). Managing the Simultaneous Execution of Coupled Phases in Concurrent Product Development, *IEEE Transaction on Management*, **43**(2): 210–217.
16. Kusiak, A. and Park, K. (1990). Concurrent Engineering: Decomposition and Scheduling of Design Activities, *International Journal of Product Research*, **28**: 1883–1900.
17. Morris, M.D. and Mitchell, T.J. (1995). Exploratory Design for Computational Experiments, *Journal of Statistical Planning and Inference*, **43**: 381–402.
18. Papalambros, P.Y. (2001). Analytical Target Cascading in Product Development, In: *Proceedings of the Third ASMO UK/ISSMO Conference on Engineering Design Optimization*, pp. 3–16, Harrogate, UK.
19. Pintér, J.D. (1996). *Global Optimization in Action*, Kluwer Academic Publishers, Dordrecht.
20. Powell, M.J.D. (2000). UOBYQA: Unconstrained Optimization by Quadratic Approximation, *Numerical Analysis Report DAMTP 2000/NA14*, University of Cambridge.
21. Sobieszczanski-Sobieski, J. and Hafka, R.T. (1996). Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments, *Presented at the 34th AIAA Aerospace Sciences Meeting and Exhibit*.
22. Stehouwer, H.P. and den Hertog, D. (1999). Simulation-based Design Optimisation: Methodology and Applications, In: *Proceedings of the First ASMO UK/ISSMO Conference on Engineering Design Optimization*, Ilkley, UK.
23. Stinstra, E.D. and den Hertog, D. (2003). *Robust Optimization Using Computer Experiments*, (Working paper).
24. Stinstra, E.D., Stehouwer, H.P. and Van Der Heijden, J. (2003). Collaborative Tube Design Optimization: An Integral Meta-modeling Approach, In: *Proceedings of the Fifth ISSMO Conference on Engineering Design Optimization*, Como, Italy.
25. Tang, D., Zheng, L. And Li, Z., et al. (2000). Re-engineering the Design Process for Concurrent Engineering, *Computers and Industrial Engineering*, **38**(4): 479–491.
26. Toropov, V.V., Filatov, A.A. and Polynkin, A.A. (1993). Multiparameter Structural Optimization Using FEM and Multipoint Explicit Approximations, *Structural Optimization*, **6**: 7–14.

Edwin van Dam

Edwin van Dam is Associate Professor at the Department of Econometrics and Operations Research at Tilburg University, the Netherlands, and a research fellow of the Royal Netherlands Academy of Arts and Sciences. He received an MSc in Mathematics at Eindhoven University of Technology in 1992, and a PhD at Tilburg University in 1996. His research interests are in Combinatorics, Design of Computer Experiments, and Simulation-based Design Optimization. Since 2001 his research is sponsored by a grant from the Royal Netherlands Academy of Arts and Sciences.

Dick den Hertog

Dick den Hertog is a Professor of Operations Research and Management Science at the Center for Economic Research (CentER), within the Faculty of Economics and Business Administration at Tilburg University, the Netherlands. He received his PhD (cum laude) in 1992 at Delft University of Technology. From 1992 until 1999 he was a consultant for optimization at CQM in Eindhoven. His research concerns Deterministic and Stochastic Simulation-based Optimization and Nonlinear Programming, with applications in engineering, production, and logistics.

Downloaded from http://cer.sagepub.com at Universiteit van Tilburg on May 14, 2007
© 2003 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.

Bart Husslage

Bart Husslage is a PhD student at the Department of Econometrics and Operations Research at Tilburg University, the Netherlands. He received an MSc in Operations Research in 2002 at Tilburg University and was awarded with the VVS-OR prize 2003 for his Master's Thesis. His areas of interest include Collaborative Optimization, Simulation-based Design Optimization, and Design and Analysis of Computer Experiments.

Peter Stehouwer

Peter Stehouwer is a Senior Consultant at Centre for Quantitative Methods (CQM) in Eindhoven, the Netherlands. He received an MSc in Computing Science in 1993 and a PhD in Mathematics at Eindhoven University of Technology in 1997. Since then, he joined CQM,where he is involved in technical optimization projects and is Program-Manager for the working area Design Optimization. His areas of interest include Operations Research in Practice, Design of Computer Experiments, Simulation-based Design Optimization, Robust Design, Metamodeling, Optimization Tools, and Neural Networks.

Erwin Stinstra

Erwin Stinstra is a consultant at Centre for Quantitative Methods (CQM) in Eindhoven, the Netherlands. He received an MSc in Applied Mathematics in 1999 from the University of Twente and is currently pursuing a PhD at the Department of Econometrics and Operations Research at Tilburg University. At CQM, he is mainly involved in technical optimization projects and is responsible for the Design Optimization Tool COMPACT. His areas of interest include practical applications of Operations Research, Design and Analysis of Computer Experiments, Simulation-based Design Optimization, Metamodeling Techniques, Robust Design and Collaborative Optimization.

Downloaded from http://cer.sagepub.com at Universiteit van Tilburg on May 14, 2007
© 2003 SAGE Publications. All rights reserved. Not for commercial use or unauthorized distribution.