Design Science

Coordination and complexity: an experiment on the effect of integration and verification in distributed design processes

Ferdinand Wöhr[1,2], Ekin Uhri[1,2], Simon Königs[1], Jakob Trauer[2],
Max Stanglmeier[1] and Markus Zimmermann[2]

[1]BMW Group, Department of Total Vehicle Development, Munich, Germany
[2]Technical University of Munich, TUM School of Engineering and Design, Department of Mechanical Engineering, Laboratory for Product Development and Lightweight Design, Garching, Germany

Abstract

The continuous integration and verification of components is essential in distributed design processes. Identifying the optimal integration and verification frequency, however, can be challenging due to the complexity of product development. Especially the effect of human decision-making in partially isolated development scenarios is difficult to consider. Thus, we performed an experimental study based on the following three steps: first, an extension of the existing *parameter design* framework, which is used to conduct experiments under laboratory conditions, in which human subjects solve quantitative surrogate design tasks. Second, a series of experiments in which 32 subjects divided into groups of two solved 229 parameter design tasks with a varying integration and verification frequency. And, third, a statistical analysis of the results with respect to development time, coupling strength and process costs. According to our results, development time can be reduced by up to 71%, if the integration and verification frequency is doubled. If process costs are also considered, the optimal frequency can be subject to a conflict of goals between reducing development time and minimising process cost.

Keywords: Distributed design, Product integration, Design processes, Parameter design

Received 11 April 2022
Revised 16 November 2022
Accepted 21 November 2022

Corresponding author
Ferdinand Wöhr
ferdinand.woehr@icloud.com

Des. Sci., vol. 9, e1
journals.cambridge.org/dsj
DOI: 10.1017/dsj.2022.26

1. Introduction

Complexity in product design poses a major challenge for companies that need to maintain their market competitiveness (Lindemann, Maurer, & Braun 2008). Developing high quality products on time may be complex for different reasons, for example, many product variants, numerous contradictory and ambiguous requirements, or strongly coupled components. Summers & Shah (2010) distinguish between complexity related to a design process, a design problem and a product itself.

In particular, products with a high degree of technical complexity, like passenger vehicles or airplanes, are normally decomposed into separate subsystems, like engines or wings, that can then be designed by multiple distributed development teams in parallel. This has two key benefits (Pimmler & Eppinger 1994). First,

https://doi.org/10.1017/dsj.2022.26 Published online by Cambridge University Press

smaller design problems with fewer design variables, fewer quantities of interest (physical measures describing the technical performance of a particular system) and fewer physical dependencies are usually easier to solve. Second, design work can be done simultaneously instead of sequentially. However, distributed design also requires *integration*, that is, the assembly of components and sub(sub)systems to a complete product realisation following a step-by-step procedure (VDI 2206 2004). Integration depends on a product's decomposition (Pimmler & Eppinger 1994), for which there often exists a variety of architectural choices.

According to Haskins *et al.* (2006), integration and verification steps are also part of the V-model – an established design methodology in product development. An illustration of the V-model with its distinct design phases is shown in Figure 1. Here, integration is about the (sequential) bottom-up aggregation of the separated product components and *verification* is about the assessment of requirements on different hierarchical levels based on the integrated components. Both processes shed light on the fulfilment of important design objectives in situations in which components evolve over time without continuous information exchange between the different parties. Especially, if isolated design decisions on lower levels affect multiple sub(sub)systems above, integration and verification provides important information about the current status of a design project and undiscovered design flaws. The left side of the V-model shows how such design flaws are then turned into revised requirements on the sub(sub)systems and components below (during requirement re-decomposition on the associated level).

During a single integration and verification step, designers perform a series of individual development activities that require close collaboration and teamwork. In early phases of product design, for example, the preprocessing of simulations performed on the system level, which can be seen as an integration, includes data collection, model setup and load case definition (Stanglmeier 2018; Stanglmeier *et al.* 2018). The postprocessing afterwards then involves an analysis, evaluation and interpretation of the results as those are compared to the given design targets. This can be seen as a verification.

Integrating subsystems and assessing their performance with respect to given design objectives usually require a tradeoff between *cost* (financial expenses and effort) and *benefit* (knowledge gained regarding a product's status). An economic analysis of this question is performed by Stanglmeier *et al.* (2018), who suggests a matrix-based framework to assess integration and verification processes in the automotive industry from a financial perspective.

Computational and analytical models, like this, are widely adopted to examine the effect of product integration and verification on different process performance metrics, such as product quality or development time. An investigation conducted by Yassine *et al.* (2003), for example, uses a so-called *work transformation model* to analyse the *design churn effect* – a dynamic phenomenon related to integration processes, in which individual participants hide design information. They reveal that, among others, feedback delays between subsystem design teams and system integrators contribute to oscillations in the project progress and, thus, play a key role when determining the stability of distributed design processes. In a different study, Mihm, Loch, & Huchzermeier (2003) propose an agent-based simulation with an 'NK' model in the background that is capable of mimicking hierarchical organisations facing coordination challenges. According to their results, hierarchy, which implies that some sort of integration and verification at the system level

https://doi.org/10.1017/dsj.2022.26 Published online by Cambridge University Press

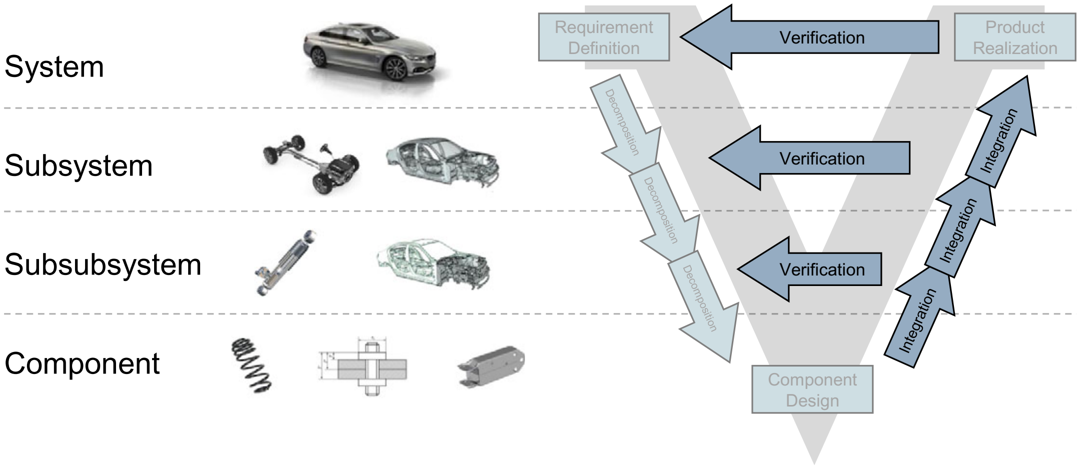

Figure 1. The V-model illustrating integration and verification steps on different hierarchical levels.

takes place, improves the search dynamics if design decisions are distributed among various designers. An agent-based model to simulate distributed design processes with a hierarchical structure is also part of the work done by Wöhr *et al.* (2020). In a parameter study, they are able to demonstrate that the rate of system integration has a considerable effect on the process dynamics. A game-theoretic approach to analyse distributed design is used by Lewis & Mistree (1997, 1998). They show that the order of design decisions influences the final product quality.

Computational methods are a powerful tool to study large-scale and complex design processes in which distributed designs need to be integrated at some point in order to verify requirements on multiple hierarchical levels. Yet, they also have a significant disadvantage: strong simplifications and assumptions that need to be made, especially in terms of human behaviour and decision-making.

An alternative approach to conduct studies in engineering design is the use of human subjects experiments in so-called *model worlds* (Szajnfarber *et al.*, 2020), where participants are supposed to solve surrogate design tasks that are abstracted from reality. Depending on the research objective, this can be done individually or in groups. In both cases, it allows a systematic analysis of scientific hypotheses under controlled conditions (Panchal & Szajnfarber 2020). The optimal timing of product integration and requirement verification, however, has not been studied with this kind of approach yet. An investigation, in particular under consideration of real human design behaviour, would, for the first time, illuminate the combined effect of process-related and human-related factors in distributed design.

Thus, in this paper, we present the results of an experimental multi-actor study in which 32 subjects in groups of 2 solved 229 parameter design problems, where the duration between each integration and verification was varied. We analyse the required completion times, the process costs and the effect of coupling strength.

Our results provide two kinds of insights: first, a better understanding of how coordination mechanisms, like integration and verification processes, affect major performance metrics (development time, cost) of product development processes. And, second, we provide a database for the calibration of simulation models.

https://doi.org/10.1017/dsj.2022.26 Published online by Cambridge University Press

The remaining paper is arranged as follows: first, literature on human subjects experiments in engineering design is reviewed and the research goal is expressed. Then, the research methodology is presented, which includes established work in parameter design, a new concept of how integration and verification processes can be combined with it, the graphical user interface (GUI) and the experimental procedure. Finally, the results are shown, analysed and discussed, and a conclusion is drawn.

2. State of the art

This literature review examines quantitative studies in engineering design, which focus on experiments where human subjects need to solve surrogate design tasks. First, we will explore research on single-actor studies, that is, where only one subject participates at a time, and then multi-actor studies, that is, where groups are involved. Beforehand, an established stage-gate type process model is used to outline what stages/phases of product development processes such experiments cover.

2.1. Parameter design task in product development

Laboratory experiments in which human subjects need to solve parametric design tasks only represent a limited period of time in product development. To illustrate that, consider the process model shown in Figure 2 (Ulrich and Eppinger, 2015).

According to this stage-gate model, product development processes can be divided into six sequential phases: (i) *Planning*: assessment of market opportunities and new technologies, (ii) *Concept Development*: identification of customer needs and specification of product concept, (iii) *System-Level Design*: definition of product architecture and decomposition of requirements, (iv) *Detail Design:* specification of component geometries and materials, (v) *Testing and Refinement:* assembly of prototypes and evaluation of performance and (vi) *Production and Ramp-Up:* product launch and start of manufacturing.

Parametric or ‘parameter’ design, where the goal is to assign proper values to some (predefined) design variables, corresponds to Detail Design and Testing and Refinement. The definition of geometries and materials during Detail Design, for example, refers to the assignment of values to the design variables by the subjects during an experiment. The assembly and testing of prototypes during Testing and Refinement, on the other hand, refers to the evaluation of a specific set of values that the subjects of an experiment have assigned to the design variables.

In earlier phases of product development, the product architecture is unknown or just about to be specified. In terms of parameter design, this would mean that the design variables themselves and the technical dependencies between them are unknown. Thus, parameter design does not represent this phase.

At the final stage of product development, the design is fully specified (fixed) and only has to be manufactured. In terms of parameter design, this would mean that the values of the design variables are ‘frozen’ and subjects cannot manipulate them anymore. Thus, parameter design does not reflect this phase either.

In summary, parameter design only represents a limited (yet important) phase of product development where the physical properties of components (geometry, material, etc.) are to be specified such that the overall design targets are satisfied.

https://doi.org/10.1017/dsj.2022.26 Published online by Cambridge University Press

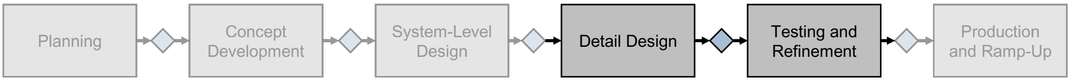

Figure 2. Generic (sequential) product development process according to Ulrich and Eppinger (2015). Parameter design represents the development work during *Detail Design* and *Testing and Refinement.*

The *signposting* simulation method proposed by Clarkson and Hamilton (2000) and Wynn *et al.* (2006) also focuses on those later phases of product development. Compared to the research approach used in this paper, however, signposting is an agent-based (computational) framework to investigate product design processes.

2.2. Literature on single-actor studies

A large amount of research focuses on the decision-making of individual subjects, that is, how they probe their design variables if the performance function is unknown and has to be explored through trial and error. The study of Borji & Itti (2013), for instance, presents an experiment in which 23 subjects are asked to identify the optimum of an arbitrary 1-D function by using experimentation. They reveal that human search is similar to Bayesian optimisation based on Gaussian processes. The impact of cost and task complexity on the decision-making of single subjects is studied by Chaudhari & Panchal (2019). Based on a similar experimental setup, they are able to show that both factors only affect the decision when to stop searching, not the actual search strategy itself. In a context-related study (where specialised domain knowledge is required in order to solve the task), Yu, de Weck, & Yang (2016) analyse the behaviour of 22 subjects who are asked to design a seawater reverse osmosis plant by using 10 key design variables. Compared to previous results, they reveal that the decision-making of top ranked subjects can be compared to a well-tuned simulated annealing algorithm, which is a meta-heuristic optimisation technique that applies exploration and exploitation. An analysis of the task completion time with respect to the task size for parametric, context-free design problems is done by Hirschi & Frey (2002). After introducing a concept called *parameter design*, the authors conduct an experimental study with 12 subjects that solve a series of coupled and uncoupled parameter design tasks with 2, 3, 4 and 5 design variables. It is shown that both the coupling and the task size have a significant effect on the task completion time. For a context-related design problem, similar results are obtained by Flager, Gerber, & Kallman (2014). In their study, subjects are asked to design a building by manipulating 2–6 design variables that are inside of a user interface which also shows a graphical representation of the object. It turns out that larger design tasks lead to a lower solution quality if the time available to solve a design task is held constant.

The effect of varying *response times*, that is, delays between the moment subjects manipulate an input variable and observe the effect, is investigated by Goodman & Spence (1978). Based on their study, in which 30 subjects are allowed to use 5 design variables in order to fit a curve into a predefined area, a response time of 1.49 seconds can already cause an increase in task completion time of 50%. This is further examined by Simpson *et al.* (2005), who, based on a set of experiments where subjects need to manipulate 2, 4 or 6 design variables to find an appropriate wing design, claim that the response time, which is varied during

https://doi.org/10.1017/dsj.2022.26 Published online by Cambridge University Press

the study, does not only affect the task completion time but also the design effectiveness (i.e., the error between the submitted design and the optimum). In their case, a 1.5 seconds response time induces a 150% error. Others, like Simpson *et al.* (2007), even note a 280% decrease in design effectiveness for the same response time. A variety of investigations, such as Simpson *et al.* (2007) or Egan *et al.* (2015), also examine the graphical representation of parametric design tasks (text-based versus graphical, static versus animated). Their conclusion is that rich GUI's increase the subjects' performance.

In summary, research on parametric single-actor studies provides an extensive body of knowledge on how individual designers act in complex design scenarios. Yet, most products are not developed by a single person but by a whole group of people who must collaborate. Studies on this issue are reviewed in the following.

2.3. Literature on multi-actor studies

It appears as if considerably less research revolves around quantitative multi-actor studies than around quantitative single-actor studies. One for our work important contribution, however, is the research undertaken by Grogan & de Weck (2016). They perform a series of experiments in which 10 groups of 3 subjects each solve 42 coupled and uncoupled parameter design tasks which are defined according to the approach suggested by Hirschi & Frey (2002). By varying the technical and social complexity, that is, the number of variables and subjects involved, it is shown that collaboration, which means the team work that is required when design tasks are solved by more than one subject, increases the completion time significantly. If three subjects are involved instead of one, for example, they note a 90% higher task completion time. A further examination of the raw data obtained by Grogan & de Weck (2016) is performed by Alelyani, Yang, & Grogan (2017). They suggest multiple regression models for subject-specific performance metrics, like the total number of design iterations or the number of design iterations during which the distance to the target area is reduced, depending on complexity and gender. The research of Austin-Breneman, Honda, & Yang (2012) deals with game-theoretic experiments in which subjects in groups of 3 have to solve a parametric satellite design task with design variables on multiple hierarchical levels. According to their results, subjects have difficulties to understand the connection between subsystems as they often falsely assume that their design decisions are divisible and independent. The authors also mention that subjects normally focus more on the individual subsystems than the system-level perspective. This, in fact, emphasises the importance of coordinated integration and verification processes which allow to assess product properties on the system level that are influenced by design variables on lower levels, such as the different sub(sub)system levels and the component level.

Instead of investigating the influence of collaboration between subjects, some studies also examine the effect of competition between multiple subjects by using parametric multi-actor design tasks, for example, Sha, Kannan, & Panchal (2014). In their work, subjects in groups of 2 compete against each other by minimising an unknown, randomly generated function that depends on a single design variable, while each trial, that is, change of the design variable, comes at a specific cost. The authors state that, for example, the cost per trial has a significant effect on how many times each subject probes its function. Similar findings are presented by Panchal, Sha, & Kanna (2017). They additionally notice that individuals shift their

https://doi.org/10.1017/dsj.2022.26 Published online by Cambridge University Press

search strategies from exploration to exploitation, which in some sense confirms that human behaviour is similar to a well-tuned simulated annealing algorithm (Yu *et al.* 2016), and that the solution subjects assume their opponent has is at least as good as the real optimum of the function. Finally, McComb, Cagan, & Kotovsky (2015) perform an experimental study in which 48 subjects in groups of 3 solve a truss design problem where some requirements at the system level are modified during the task. Based on their results, successful groups tend to use different problem-solving techniques in a sense that they select more simple designs and search focused areas of the design space.

3. Research objective

As shown, literature on single- and multi-actor studies in engineering design covers a wide range of topics like human decision-making, collaboration, team dynamics or competition. Coordination procedures, like integration and verification phases, which are predefined in large-scale companies in order to synchronise distributed design teams, have not been analysed regarding their effect on development time and process cost. This has led us to define the following research question:

> What is the relative effect of a varying time interval between each integration and verification on development time and cost in case of small design problems (that are represented by surrogate design tasks)?

We deliberately focus on small-scale design problems since they allow a better understanding of the key mechanisms. It is possible that these effects also drive the dynamics of real-world development processes.

Any answers obtained would be beneficial from a scientific point of view – but also from an industrial perspective, since many organisations apply an integration and verification strategy where the complete product is assembled and examined in predefined time intervals, see Stanglmeier (2018) and Stanglmeier *et al.* (2018), without being aware of what longer or shorter time spans would actually mean. In fact, a ‘flexible’ integration of subsystems without fixed intervals, for example, depending on the individual progress, could provide an additional alternative to improve the efficiency of distributed design processes. Unfortunately, however, a quantitative relationship between the integration and verification frequency and development time is not known, which underlines the relevance of our research question.

4. Research method

Our scientific approach is based on the parameter design framework (Hirschi & Frey 2002; Grogan & de Weck 2016). We extend the theoretical foundation as well as the software implementation of this framework to account for integration and verification phases in distributed design processes.

4.1. Established groundwork in parameter design

Parameter design is a quantitative approach to investigate development processes based on surrogate design tasks that are solved either by individuals or groups of

https://doi.org/10.1017/dsj.2022.26 Published online by Cambridge University Press

subjects, while the required completion time is captured and evaluated regarding different process performance metrics. Our notation is adopted from Grogan & de Weck (2016). In each surrogate design task, a set of input variables, denoted as $\mathbf{x}$, can be altered in order to manipulate a set of output variables, denoted as $\mathbf{y}$, for which design goals (requirements) are specified. The mapping between input and output variables, that is, the technical dependencies between them, is defined by the coupling matrix $\mathbf{M}$. Based on this, each system model can be described as:

$$\mathbf{y} = \mathbf{M}\mathbf{x}. \tag{1}$$

As in previous studies (Hirschi & Frey 2002; Grogan & de Weck 2016), we assume linear dependencies between the input and output variables (note that this is a strong assumption as real-world systems are often nonlinear). If the number of input variables is N and the number of output variables is K, Eq. (1) becomes

$$\mathbf{y} = \mathbf{M}\mathbf{x} \Rightarrow \begin{bmatrix} y_1 \\ \vdots \\ y_K \end{bmatrix} = \begin{bmatrix} m_{11} & \cdots & m_{1N} \\ \vdots & \ddots & \vdots \\ m_{K1} & \cdots & m_{KN} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}. \tag{2}$$

For simplicity, our analysis is restricted to $N = K$ (i.e., same number of input and output variables), which, as before, is in line with the procedure of former studies. In uncoupled design tasks, only the diagonal elements of $\mathbf{M}$ are nonzero numbers. This means that each output variable y_i only depends on the corresponding input variable x_j and the dependency m_{ij}. Fully coupled systems, on the other hand, are characterised by coupling matrices $\mathbf{M}$ that include nonzero elements only, which means that $m_{ij} \neq 0 \forall i, j$.

Tasks in parameter design are usually defined context free, which means that no domain-specific expertise or expert know-how is needed in order to solve them. This is essential for experimental studies in which the effect of special knowledge shall be excluded, like in this paper.

An important measure to characterise the coupling strength of a system is the design matrix trace, which, according to Hirschi (2000), compares the magnitude of all diagonal elements to the Euclidean norm of all elements of the coupling matrix if $N = K$ (in our case, the coupling strength lies between 0.11 and 1.41):

$$t(\mathbf{M}) = \frac{\sum_{i=1}^{N} |m_{ii}|}{\sqrt{\sum_{i,j=1}^{N} m_{ij}^2}}. \tag{3}$$

Note that this definition is not the same as the established mathematical definition of a trace. In the following, any use of this term refers to Eq. (3).

The target values assigned to all output variables are denoted as $\mathbf{y}^\star$. A design task is considered to be solved if the error function E, which we assume to be the distance between each output variable and the corresponding target value, which means

$$\mathrm{E}_i = \left\{ |y_i - y_i^\star| \right\}, \tag{4}$$

is less than or equal to the given error tolerance $\mathbf{E}^\star$, meaning that:

$$\mathbf{E}(\mathbf{y}, \mathbf{y}^\star) \leq \mathbf{E}^\star. \tag{5}$$

https://doi.org/10.1017/dsj.2022.26 Published online by Cambridge University Press

By assigning an identical error tolerance to all output variables, that is, $\mathrm{E}_i^\star = \varepsilon$, the problem statement for each surrogate design task can be formulated as:

$$\text{find}\,\mathbf{x},\ s.t. \left| \sum_j m_{ij} x_j - y_i^\star \right| \leq \varepsilon, \forall i. \tag{6}$$

In single-actor design scenarios, all input and output variables are assigned to the same subject. This means one actor *controls* all x_j and, at the same time, *monitors* the influence on all y_i. In case of multi-actor design scenarios, however, input and output variables are distributed among multiple subjects. Thus, one actor might observe a change in its output variable y_i, which is caused by a change of an input variable x_j that is controlled by a different actor. To formalise the assignment of input and output variables, two binary matrices, **I** and **O**, can be used. The matrix **I**, with the dimension $n \times N$, where n represents the number of subjects, describes the mapping of subjects to input variables. Each entry I_{sj} is defined as:

$$I_{sj} = \begin{cases} 1, & \text{if output}\, j \text{ is assigned to subject}\, s, \\ 0, & \text{else.} \end{cases} \tag{7}$$

In a similar way, the matrix **O**, with the dimension $n \times K$, represents the mapping of subjects to output variables. Each entry O_{si} is defined as:

$$O_{si} = \begin{cases} 1, & \text{if input}\, i \text{ is assigned to subject}\, s, \\ 0, & \text{else.} \end{cases} \tag{8}$$

4.2. Modelling integration and verification phases

In previous studies with a focus on multi-actor parameter design, such as the one from Grogan & de Weck (2016), the flow of information between subjects was assumed to be instantaneous. This means that any change in an input variable was directly mapped onto all output variables. Real-world product design processes, however, are usually characterised by phases in which stakeholders work isolated from each other, without constant feedback on design changes until an integration and verification takes place. This means that the information flow across domain interfaces is blocked for some time during which stakeholders potentially rely on obsolete design information.

In the automotive industry, for example, subsystems, like the engine, gear-box or body, are developed by distributed design teams, who eventually do not receive any design update from others for several weeks or even months. At some point, these subsystems are then geometrically assembled and used as input for different system simulations based on the finite element method (FEM), computational fluid dynamics (CFD) or multibody simulation (MBS), see Wöhr *et al.* (2020). This is an integration. Once the simulations are completed, the results are compared to the given requirements (e.g., some crash performance). This is a verification.

In order to incorporate these three phases, we extend the existing framework by dividing each parameter design experiment into three phases that are repeated periodically: (i) *isolated design*, (ii) *integration* and (iii) *verification.*

During *isolated design*, subjects manipulate their input variables to meet the design targets (requirements + error tolerances) of their output variables, without exchanging design information with other subjects. This means that, first, design

https://doi.org/10.1017/dsj.2022.26 Published online by Cambridge University Press

changes related to the own input variables do not affect output variables assigned to others, even if they depend on it. And, second, design changes related to input variables that are assigned to other subjects do not affect the own output variables, even if they depend on it. In essence, information flow across domain interfaces is blocked during that time. To formalise this, we suggest a mathematical extension of the parameter design framework: for each subject there is a local representation of the entire system model, consisting of $\mathbf{x}^s = \left[x_1^s, \ldots, x_N^s\right]$ and $\mathbf{y}^s = \left[y_1^s, \ldots, y_K^s\right]$. Just as before, $\mathbf{x}^s$ and $\mathbf{y}^s$ are related by:

$$\mathbf{y}^s = \mathbf{M}\mathbf{x}^s. \tag{9}$$

Within $\mathbf{x}^s$ and $\mathbf{y}^s$, a subject still only controls the input variables and monitors the output variables that he or she is assigned to according to **I** and **O**. The remaining entries are interim values that originate from the last integration and verification.

During *integration*, the latest status of all input variables is assembled based on each local representation of $\mathbf{x}$ and the assignment of subjects to input variables stored in **I**. First, each local representation is preconditioned (i.e., turned into $\tilde{\mathbf{x}}^s$), such that only the entries a subject is responsible for remain:

$$\tilde{x}_j^s = \begin{cases} x_j^s & \text{if } I_{sj} = 1, \\ 0 & \text{if } I_{sj} = 0. \end{cases} \tag{10}$$

The most recent design, denoted as $\mathbf{x}'$, is identified based on all $\tilde{\mathbf{x}}^s$:

$$\mathbf{x}' = \sum_{s=1}^{n} \tilde{\mathbf{x}}^s. \tag{11}$$

The result $\mathbf{x}'$ is then used to compute the most recent system performance $\mathbf{y}'$, that is, the current status of all output variables, based on Eq. (1).

During *verification*, the most recent system performance is then compared to the design targets, see Eqs. (4) and (5), in order to evaluate whether the design task is solved. If this is the case, the experiment is terminated. If it is not the case, each local representation of $\mathbf{x}$ and $\mathbf{y}$ is updated based on the most recent design ($\mathbf{x}'$) and the most recent system performance ($\mathbf{y}'$), which means that:

$$\{\mathbf{x}^s = \mathbf{x}'\}\forall s, \tag{12}$$

and:

$$\{\mathbf{y}^s = \mathbf{y}'\}\forall s. \tag{13}$$

After this, the next isolated design phase begins.

Note that subjects are not actively involved during integration and verification, that is, the computations during both phases are performed automatically without any interference or intervention. In reality, this might be quite different, as integration and verification phases usually require intense human effort and additional labour in order to process all of the design information (see description of detailed design work during each integration and verification step in introduction of this paper). For simplicity, we neglect the specific activities performed during both phases.

Each phase, that is, isolated design, integration and verification, has a predefined duration: t_{iso},t_{int} and t_{ver}. In our case, both the integration and verification occur instantaneously (i.e., $t_{int} = t_{ver} = 0s$). The duration of isolated design, on the

https://doi.org/10.1017/dsj.2022.26 Published online by Cambridge University Press

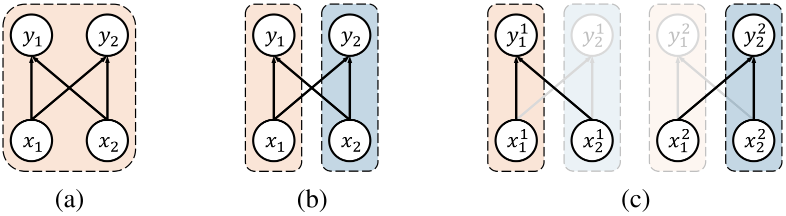

Figure 3. Attribute dependency graphs (ADGs) representing different design tasks which are used in: (a) Hirschi & Frey (2002), (b) Grogan & de Weck (2016) and (c) this work.

other hand, is varied between 0, 3, 6, 9 and 12 seconds. Note that in case of 0 seconds, each design change, that is, update of $\mathbf{x}^s$ and $\mathbf{y}^s$, performed by any of the subjects is automatically followed by an integration and verification phase. This means each local design modification triggers a system assembly and requirement evaluation. The setup $t_{iso} = t_{int} = t_{ver} = 0s$ corresponds exactly to the information exchange conditions used by Grogan & de Weck (2016). We associate lower values of t_{iso} with higher integration and verification frequencies (and vice versa).

Surrogate design tasks and the assignment of variables can be visualised with *attribute dependency graphs* (or ADGs), see (Zimmermann *et al.* 2017; Rötzer *et al.* 2022). According to this concept, physical properties of a technical system, such as the input and output variables in parameter design, are shown as vertices and the dependencies between them, such as the linear functions that we assume, are shown as edges. The colours reveal who controls and monitors which variable. Figure 3a, for example, shows a simple *single-actor* design task with two input and two output variables which are controlled and monitored by the same subject. This is representative for the research of Hirschi & Frey (2002). A *multi-actor* design task with two input variables and two output variables that are split among multiple subjects is depicted in Figure 3b. This is representative for the research of Grogan & de Weck (2016). By contrast, the setting of this work is visualised in Figure 3c. In this case, it is not possible to represent a design task by using a single ADG because the current state of design is controlled by various subjects who (partially) work isolated from each other.

4.3. Software implementation and user interface

Our new approach (for the investigation of integration and verification processes) is integrated into an existing open-source software devised by Grogan (2019). In its original state, the software, which contains a front-end and back-end, supports parameter design experiments that are focused on single- and multi-actor settings with varying degrees of technical and social complexity. For this study, we added an algorithm to the back-end which organises the three recurring phases: isolated design, integration and verification. It is set up in a way that allows the duration of isolated design to be varied independently.

The unmodified front-end, or GUI, of the software architecture for a subject who controls one input variable and monitors one output variable is depicted in Figure 4. It contains two sliders: a vertical one for the input variable and a horizontal one for the output variable. The relative position of both equals the

https://doi.org/10.1017/dsj.2022.26 Published online by Cambridge University Press

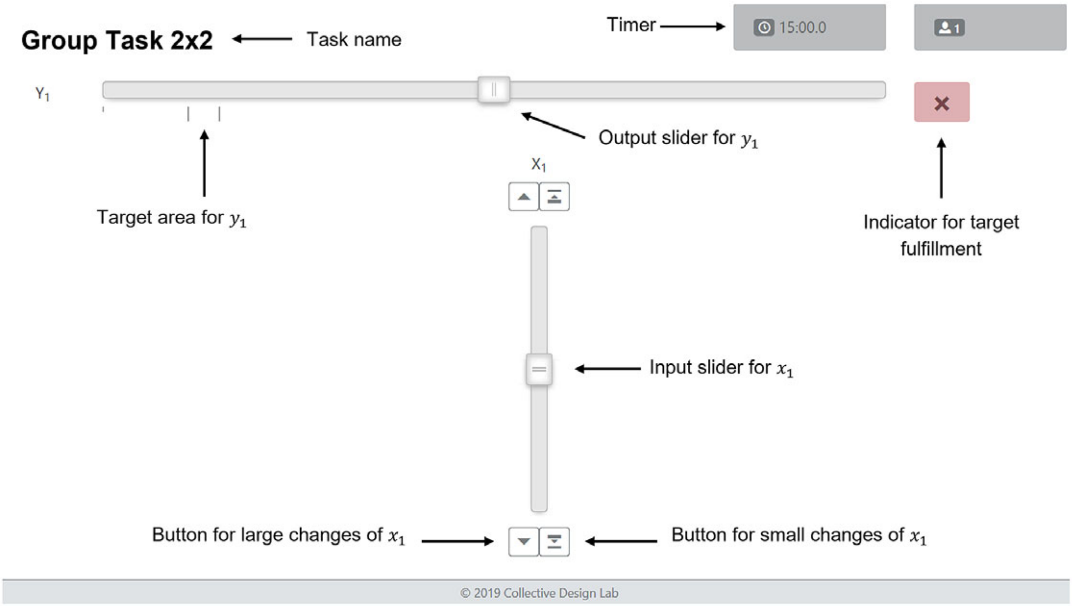
Group Task 2x2
Task name
Timer
15:00.0
1
Y_1
Output slider for y_1
Target area for y_1
X_1
Indicator for target fulfillment
Input slider for x_1
Button for large changes of x_1
Button for small changes of x_1
© 2019 Collective Design Lab

Figure 4. Graphical user interface of the open-source software from Grogan (2019).

status of $\mathbf{x}^s$ and $\mathbf{y}^s$. Adjusting the output slider (during isolated design) is only possible by manipulating the input slider which, in turn, can be controlled by drag-and-drop or by clicking the buttons on the top or bottom. As in previous studies, changes made to the input slider (during isolated design) are instantly mapped onto the output slider (of the own GUI) without a delay. Two small lines below the output slider show the target area. If a subject reaches that area during a multi-actor experiment, however, it does not automatically mean that the overall design task is solved, as others may still try to reach their design goals. Yet, those might be unachievable due to the latest input variables they received. Whether or not the local design goals of a subject are currently satisfied is shown by a green tick or a red cross on the right-hand side of the interface. A maximum duration is specified for each design task after which an experiment is terminated. The time remaining to solve a design task is displayed in the upper right corner. Note that each subject is only permitted to observe his or her own GUI. The aim of this is to simulate the communication barriers and limited information exchange inherent to distributed design processes.

During isolated design, only the subjects themselves can cause a movement of their output slider by manipulating their input slider. This allows them to probe different designs in order to anticipate the dependency between $\mathbf{x}^s$ and $\mathbf{y}^s$ without receiving constant feedback (possible disturbance) resulting from design changes made by others. During each integration phase, the output sliders of all subjects are then updated based on the latest $\mathbf{y}'$. The new positions of all output sliders are then compared to the target intervals during each verification phase. Again, note that both of these phases occur instantaneously and the subjects are not involved. It can be assumed that time delays caused by network latencies are small and so have no impact on the results.

https://doi.org/10.1017/dsj.2022.26 Published online by Cambridge University Press

In an attempt to establish the same boundary conditions as in previous studies (see Hirschi & Frey (2002) as well as Grogan & de Weck (2016)) subjects are not informed about the dependencies between the input and output variables ($\mathbf{M}$), the assignment of subjects to input and output variables ($\mathbf{I}$, $\mathbf{O}$) and the numeric values of the own input and output variables during an experiment. The GUI includes all the information that subjects are aware of in the course of an experiment. The aim of this is to prevent any reverse engineering that might allow subjects to solve design tasks analytically.

4.4. Design of experiments and procedural setup

Selection of design tasks

Based on the theoretical foundation, we conduct two series of experiments: first, a *single-actor study* with the goal to replicate previous results obtained by Hirschi & Frey (2002) and Grogan & de Weck (2016). This step allows us to confirm earlier findings and, thus, approve the reliability of our setup. As in both of those studies, we use design tasks with 2, 3 and 4 input and output variables that are all assigned to the same subject. This can be expressed as $n = 1$ (number of subjects) and $2 \leqslant N, K \leqslant 4$ (number of input and output variables). All of the three design scenarios in the first experimental series, for which the original parameter design framework and initial open-source software is used, are shown in Figure 5a–c. In a second *multi-actor study*, we then analyse the effect of varying time intervals between each integration and verification by using design tasks with two input variables, two output variables ($N, K = 2$) and two subjects ($n = 2$). Each subject is responsible for one of the input and output variables. This scenario is illustrated in Figure 5d. At this point, we apply the extended parameter design framework and the adjusted open-source software outlined above to vary the time span between each integration and verification between 0, 3, 6, 9 and 12 seconds.

Design task generation

In order to be consistent with the practice of Grogan & de Weck (2016), we use the same concept to generate design tasks, which implies randomising $\mathbf{M}$ and $\mathbf{y}^{\star}$. With respect to the coupling matrix $\mathbf{M}$, this means establishing orthonormal bases of vectors, whereby the entries are chosen from a uniform distribution $(0,\ 1)$ while assuming $m_{i,j} \in [-1;1]$. This ensures an appropriate relation between input and output variables and guarantees uniquely determined design tasks with one exact solution. The target vectors $\mathbf{y}^{\star}$ are also established by composing an orthonormal basis of vectors that are drawn from a uniform distribution $(0, 1)$; here the entries are $y_i^{\star} \in [-1;1]$ and the Euclidean norm of each vector is $\|\mathbf{y}^{\star}\| = 1$. This setup ensures a standardised distance to the solution for all design tasks. Furthermore, we define the range of possible inputs as $x_j \in [-1;1]$ and the initial conditions as $\mathbf{x_0} = \mathbf{y_0} = \mathbf{0}$. It is guaranteed that the starting position is not directly an acceptable solution. To fully comply with the study of Grogan & de Weck (2016), we use an error tolerance of $\varepsilon = 0.05$ and choose surrogate design tasks which are fully coupled, that is, $m_{ij} \neq 0 \forall i,j$.

https://doi.org/10.1017/dsj.2022.26 Published online by Cambridge University Press

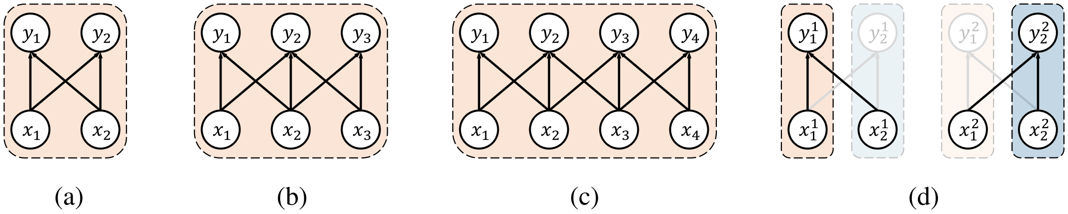

Figure 5. Attribute dependency graphs (ADGs) representing different design tasks in our experimental study: (a), (b), (c) single-actor study and (d) multi-actor study.

Subjects and procedure

A total of 34 subjects from an automotive manufacturing company participated in the experiments, which were conducted virtually via an online communication platform. All subjects, whose demographic data are shown in Table 1, received an email containing basic information about the study together with an invitation to take part on a voluntary basis. None of them received compensation. All subjects were divided into groups of 2 and then assigned to a 90 minute time slot (session), arranged based on their availability. At the beginning of each session, each group was introduced to the user interface (GUI), familiarised with the objective of the study and informed about the rules. Rules contained two major requests: first, not to use any external tool, such as pen and paper, or a calculator (even if numeric values were not displayed) to further reduce the risk of reverse engineering and, second, not to engage in any form of communication, for example, talking or messaging, since we observed in a preliminary study that conversations among subjects can affect the dynamics of experiments considerably, especially if individual subjects dominate others and attempt to coordinate the decision process. Finally, multiple training rounds were performed to enable the subjects to become accustomed to the experimental setup and the given rules.

Then, the main part of each session began. This included a set of single-actor experiments done individually and simultaneously, along with a set of multi-actor experiments, done in pairs. A list of all design tasks presented to the subjects is given in Table 2. In both studies, the order of tasks was randomised to prevent any learning or adaptation of behaviour during the experiments. There were, however, two exceptions: the 4 × 4 design tasks were not presented at the beginning of the single-actor study, and the design tasks with 12 seconds between each integration and verification were not presented at the beginning of the multi-actor study. This was done in order not to overstress, confuse or demotivate the subjects by giving them a challenging task right away. It can be assumed that these exceptions had a negligible effect and that the randomisation negated any influences that might have been caused by the order of the design tasks. Each set of repetitions (three per task size and three per interval) was performed together.

Before each experiment in the multi-actor study, the subjects were informed about the current time interval between each integration and verification to enable them to anticipate when design information is to be shared (like in reality). After each session, a qualitative evaluation sheet was sent to the subjects, in which they could give feedback and explain their decision-making.

https://doi.org/10.1017/dsj.2022.26 Published online by Cambridge University Press

https://doi.org/10.1017/dsj.2022.26 Published online by Cambridge University Press

Table 1. Demographic data of the subjects who took part in the single- and multi-actor study. Data was reported by the subjects themselves. Note that only 32 of the 34 subjects participated in the multi-actor study

Category	Option	Number	Percentage	Category	Option	Number	Percentage
Overall		34	100	Employment status	Intern	11	32.35
Industry	Automotive	34	100		Writing thesis	2	5.88
Gender	Male	26	76.47		Working student	1	2.94
	Female	8	24.24		PhD candidate	1	2.94
Age	18–24	12	35.29		Senior employee	19	55.88
	25–29	6	17.65	Work experience in a technical field	None	9	26.47
	30–34	7	20.59		1–2 years	6	17.65
	35–39	4	11.76		3–4 years	8	23.53
	40–44	2	5.88		5–6 years	4	11.76
	45+	3	8.82		7+ years	7	20.59
Nationality	German	29	85.29	Frequency of interaction with other subject	Never	17	50
	Other	5	14.71		Once	1	2.94
Education	High school	8	23.53		Rarely	8	23.53
	Bachelor’s degree	9	26.47		Monthly	2	5.88
	Master’s degree	12	35.29		Weekly	3	8.82
	Doctoral degree	5	14.71		Daily	3	8.82

Table 2. Experimental procedure of the single- and multi-actor study. 2 × 2 design tasks have 2 input variables and 2 output variables, 3 × 3 design tasks have 3 input variables and 3 output variables and 4 × 4 design tasks have 4 input variables and 4 output variables

Single-actor study			Multi-actor study		
Task size	Repetitions	Max. time	t_{iso}	Repetitions	Max. time
2 × 2	3 per subject	2 minutes	0 seconds	3 per pair	15 minutes
3 × 3	3 per subject	4 minutes	3 seconds	3 per pair	15 minutes
4 × 4	3 per subject	10 minutes	6 seconds	3 per pair	15 minutes
			9 seconds	3 per pair	15 minutes
			12 seconds	3 per pair	15 minutes

5. Results

This section presents the results of the single- and multi-actor study by illustrating the data graphically and describing the observed effects.

We also employ a number of statistical methods to compare our results with those of Grogan & de Weck (2016) (single-actor study) and to evaluate whether the time interval between each integration and verification has a significant effect on the task completion time (multi-actor study).

5.1. Single-actor study

Table 3 shows the descriptive data of the single-actor study regarding the effect of different task sizes on the task completion time compared to the results obtained by Grogan & de Weck (2016). Of the 291 design tasks shown to the subjects, 8 have been omitted due to connection issues and time overruns. Thus, 283 samples remain.

Graphical representation

Figure 6 shows the results of the single-actor study compared to the data obtained by Grogan & de Weck (2016). The size of a box represents the interquartile range (IQR), which is defined as the distance between the first quartile Q_1 (median of the lower half of a data set) and the third quartile Q_3 (median of the upper half of a data set), that is, $IQR = Q_3 - Q_1$. The lower and upper whisker represent the minimum and maximum values of each data set, that is, $Q_1 - 1.5IQR$ and $Q_3 + 1.5IQR$, if the outlier (shown as plus signs) are excluded. The horizontal bar within each box illustrates the median.

Description of results

The data indicates that both experimental series match each other well not only when comparing them regarding a specific task size but also when analysing their trend regarding an increase in task size. In general, larger design tasks have two significant consequences: first, on average, more time is needed in order to solve them as the mean and median increase exponentially with the task size and, second, the distance between the minimum and maximum times needed increases along

https://doi.org/10.1017/dsj.2022.26 Published online by Cambridge University Press

Table 3. Statistical properties of task completion times obtained for the single-actor study compared to Grogan & de Weck (2016)

	Our results			Grogan & de Weck (2016)		
Task size	2 × 2	3 × 3	4 × 4	2 × 2	3 × 3	4 × 4
Min (seconds)	5.38	8.68	24.18	3.76	5.77	18.29
Q_1 (seconds)	9.77	29.05	62.66	8.96	29.13	59.92
Median (seconds)	12.96	42.08	92.24	14.36	39.95	110.20
Q_3 (seconds)	19.49	64.64	150.33	20.81	59.05	151.31
Max (seconds)	38.44	188.86	527.33	39.98	201.94	368.03
Mean (seconds)	15.36	53.40	121.10	15.85	56.09	118.29
Std. dev. (seconds)	7.85	36.74	90.91	8.20	44.50	78.74
Total tasks	99	101	91	60	60	60
Omitted tasks	0	3	5	9	9	19

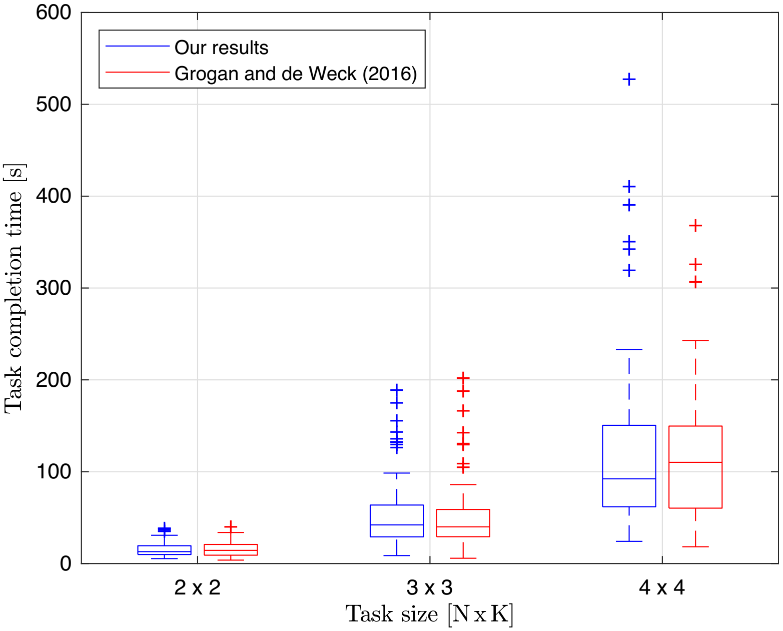

Figure 6. Results of the single-actor study compared to Grogan & de Weck (2016).

with the number of design variables as the IQR and the distance between the lower and upper whiskers increase. In fact, compared to design tasks with two input and two output variables, design tasks with three and four input and output variables also display more outliers.

https://doi.org/10.1017/dsj.2022.26 Published online by Cambridge University Press

Statistical analysis

As the correlation between task size and task completion time is already well known (Hirschi & Frey 2002; Grogan & de Weck 2016), we do not attempt to prove a significant relationship between these two variables. To evaluate the reliability and repeatability of our setup, we apply a statistical test which compares our data to the data obtained by Grogan & de Weck (2016) for the 2 × 2, 3 × 3 and 4 × 4 task sizes. Therefore, it is necessary to analyse whether or not the associated data is normally distributed. A Shapiro–Wilk test (Shapiro & Wilk 1965; Royston 1992) performed on each sample (see Table 4) reveals that the p-value is considerably smaller than the given significance level. Hence, our hypothesis (data is normally distributed) can be rejected for every sample.

Based on the results of the Shapiro–Wilk test and the fact that the two samples are independent for each task size, the Mann–Whitney U test (Mann & Whitney 1947) can be applied in order to evaluate whether there is a significant difference between each of these pairs. The results of the test for each task size are presented in Table 5. It can be seen that in all three cases the resulting p-value is larger than the given significance level of 0.05. This means that our hypothesis for each task size (no significant difference between both samples) must be retained.

In summary, both the visual inspection and the statistical analysis show that there is no significant difference between our results and those of Grogan & de Weck (2016) for the single-actor study.

5.2. Multi-actor study

The experimental results of the multi-actor study are analysed in three ways: first, the time interval between each integration and verification is assessed regarding its effect on the task completion time. Then, the coupling strength between actors is analysed on whether it influences the occurrence of high task completion times. And finally, the time interval between each integration and verification is studied regarding its simultaneous effect on the task completion time and process cost.

Effect of time interval between each integration and verification

Table 6 presents the descriptive statistics of the multi-actor study when comparing different time intervals between each integration and verification in terms of their effect on the task completion time. Of the 229 design tasks shown to the subjects, 4 have been omitted due to connection issues and time overruns. Figure 7 shows the associated data graphically based on the same type of diagram used before in the single-actor study.

Description of results. Increasing the time span between each integration and verification has multiple effects on the task completion time. For one, the average time required to solve a design task increases continuously as can be seen by the median values between 0 and 12 seconds. Doubling the time span from 3 to 6 seconds, or from 6 to 9 seconds, results in a 40–71% increase in task completion time. Furthermore, the difference between the shortest and the longest time required to find a good design can also be seen to increase when observing the range between 0 and 6 seconds. This is evidenced by the growth of both the box and the distance between the two whiskers. However, between 6 and 12 seconds, this trend cannot be confirmed. Instead, the variance of the distribution remains

https://doi.org/10.1017/dsj.2022.26 Published online by Cambridge University Press

Table 4. Shapiro–Wilk test (Shapiro & Wilk 1965; Royston 1992) for the single-actor study, with a significance level of $\alpha = 0.05$

	Our results			Grogan & de Weck (2016)		
Task size	2 × 2	3 × 3	4 × 4	2 × 2	3 × 3	4 × 4
df	99	98	86	51	51	41
W_{stat}	0.8919	0.8355	0.7762	0.9405	0.7889	0.8638
p-value	<0.001	<0.001	<0.001	<0.015	<0.001	<0.001

Table 5. Mann–Whitney U test (Mann & Whitney 1947) for the single-actor study, with a significance level of $\alpha = 0.05$

Task size	U_{min}	μ_U	σ_U	z-value	p-value
2 × 2	2330	2475	249	–0.583	0.56
3 × 3	2423	2474	248	–0.204	0.83
4 × 4	1752	1763	194	–0.058	0.95

Table 6. Statistical properties of task completion times obtained in multi-actor study

	Time interval between each integration and verification (t_{iso})				
	0 seconds	3 seconds	6 seconds	9 seconds	12 seconds
Min (seconds)	5.27	9.00	12.00	27.00	12.00
Q_1 (seconds)	10.42	18.00	30.00	49.50	48.00
Median (seconds)	14.94	30.00	42.00	72.00	84.00
Q_3 (seconds)	28.81	73.50	111.00	126.00	120.00
Max (seconds)	697.03	174.00	444.00	621.00	364.00
Mean (seconds)	50.74	51.47	81.87	109.00	98.05
Std. dev. (seconds)	119.39	46.48	86.06	111.42	67.05
Total tasks	48	45	45	47	44
Omitted tasks	1	0	0	2	1

roughly the same and between 6 and 9 seconds, it even decreases. Between 6 and 12 seconds, the vertical position of the box stabilises and between 9 to 12 seconds, it even drops slightly even though the median continues to increase, as stated above. The duration between each integration and verification does not seem to effect the number and intensity of outliers.

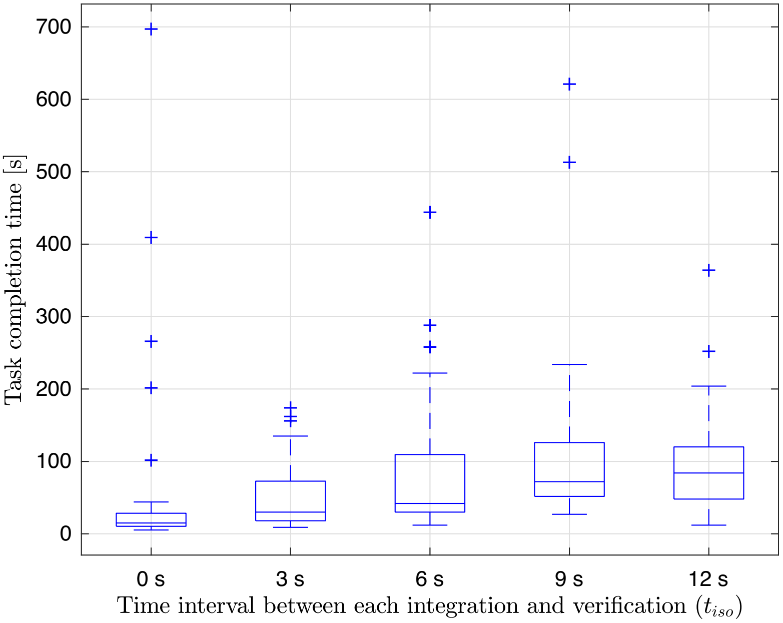

Figure 7. Results of the multi-actor study.

Statistical analysis. Applying a statistical test to prove that the time intervals between each integration and verification significantly affect the task completion time again depends on whether the corresponding data is normally distributed. A Shapiro–Wilk test is performed on each sample, that is, for all task completion times obtained for a specific time interval between each integration and verification (see Table 7). It can be seen that the p-value for all samples is below the significance level, that is, the critical threshold. This means that our hypothesis, that is, that the data obtained is normally distributed, must be rejected for each sample.

According to the outcome of the five Shapiro–Wilk tests and the circumstance that all of the samples are dependent, a Skillings–Mack test (Skillings & Mack 1981; Chatfeld & Mander 2009) is used to assess whether there is a significant relationship between the time span between each integration and verification and the task completion time. This statistical test is an extension of the more familiar Friedman test. It is used when data is missing, whether by design or by accident. In our case, the test results in a p-value of < 0.001 with $\chi^2(4) = 73.97$, which is considerably smaller than the significance level of 0.05. This means that there is a significant relationship between the time interval between each integration and verification and the task completion time.

Effect of technical coupling strength between subjects

As mentioned previously, the aim of generating design tasks based on Grogan & de Weck (2016) is to create a randomised (individual) coupling matrix and target vector for each experiment. This ensures that the findings are valid for any linear set of equations with a balanced relationship between input and output variables, and not just for one specific design task. When assigning technical responsibility to

https://doi.org/10.1017/dsj.2022.26 Published online by Cambridge University Press

Table 7. Shapiro–Wilk test (Shapiro & Wilk 1965; Royston 1992) for the multi-actor study, with a significance level of $\alpha = 0.05$

	Time interval between each integration and verification (t_{iso})				
	0 seconds	3 seconds	6 seconds	9 seconds	12 seconds
df	47	45	45	45	43
W_{stat}	0.3654	0.8008	0.7077	0.5887	0.8312
p-value	<0.001	<0.001	<0.001	<0.001	<0.001

different subjects (see **I** and **O**), however, randomised coupling matrices result in a different setting each time regarding how strongly the design decisions of one subject (variation in input variable) influence the system performance of another subject (variation in output variable).

To illustrate this, consider the 2 × 2 system model used in the multi-actor study with the coupling matrix **M**, the output variables y_1 (assigned to subject A) and y_2 (assigned to subject B) and the input variables x_1 (assigned to subject A) and x_2 (assigned to subject B):

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \tag{14}$$

Here, the randomly defined entries of the coupling matrix determine how strongly a change made by subject A to x_1 influences subject B's variation of y_2 and how strongly a change made by subject A to x_1 influences its own output variable, and vice versa.

The (design matrix) trace $t(\mathbf{M})$ is an essential quantity for characterising the coupling strength of a system. In our case, it also measures how strongly subjects influence their own output variable compared to the overall dependency strength, see Eq. (3). Applied to Eq. (14), the trace can be written as:

$$t(\mathbf{M}) = \frac{|m_{11}| + |m_{22}|}{\sqrt{m_{11}^2 + m_{12}^2 + m_{21}^2 + m_{22}^2}}. \tag{15}$$

A large trace indicates that both subjects influence their own output variable more than they do the output variable of the other subject. A small trace, in comparison, indicates that both subjects influence the output variable of the other subject more than their own one. In this study, the trace therefore represents relative measure, which compares the dependencies within an area of responsibility to the sum of all the dependencies at play.

We will now examine all task completion times of the multi-actor study under consideration of all trace values that can be determined for each experiment based on the corresponding coupling matrices. Again, note that the trace values are not directly controlled like an independent variable but result from the generation of design tasks according to the procedure of Grogan & de Weck (2016). Figure 8 shows the associated data.

https://doi.org/10.1017/dsj.2022.26 Published online by Cambridge University Press

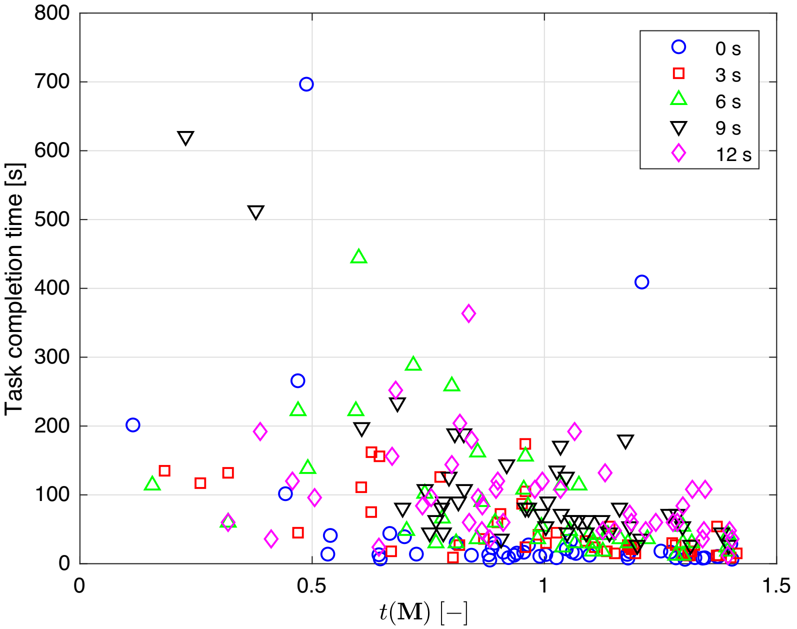

Figure 8. Results of the multi-actor study evaluated with respect to coupling strength and task completion time for different time intervals between each integration and verification.

Description of results. Overall, lower values of the design matrix trace cause higher task completion times as, for example, 14 design tasks with a trace of less than one, that is, $t(\mathbf{M}) < 1$, are completed in over 200 seconds, whereas only 1 design task with a trace of more than one, that is, $t(\mathbf{M}) > 1$, is solved in over 200 seconds (which corresponds to a ratio of 93% to 7%). Furthermore, 16 of the 18 (88.89%) design tasks that are either outliers or not completed have a trace of less than one. Hence, the task completion time changes considerably depending on the coupling strength of a system, as can be seen when assessing the data as a whole. Now, we will analyse the different time intervals between each integration and verification separately. This allows two major observations. First, for time intervals of 3, 6, 9 and 12 seconds, the average task completion time and the statistical variance increase at a roughly constant rate, if the trace is reduced. Second, for time intervals of 0 seconds, the task completion time remains approximately constant between $t(\mathbf{M}) \approx 0.7$ and $t(\mathbf{M}) \approx 1.4$ and increases between $t(\mathbf{M}) \approx 0.7$ and $t(\mathbf{M}) \approx 0.1$.

Relationship between task completion time and theoretical cost

Different time intervals between each integration and verification might affect not only the task completion time but also other metrics that are used to characterise the performance of development processes. Here, we specifically analyse the cost that would be incurred if subsystems are integrated and verified more frequently.

In order to determine the theoretical process cost of an experiment, we assign a nondimensional cost factor to each integration and verification (c_{iv}) and to each iteration (c_{it}), by which we mean each design change (update of $\mathbf{x}^s$, $\mathbf{y}^s$) performed

https://doi.org/10.1017/dsj.2022.26 Published online by Cambridge University Press

by any of the subjects. Based on the number of integrations and verifications (n_{iv}) and the number of iterations (n_{it}), both of which are tracked during an experiment and a generic (linear) cost model suggested by Wöhr *et al.* (2020), the total costs (c_{tot}) can be determined as follows:

$$c_{tot} = n_{it}\,c_{it} + n_{iv}\,c_{iv} \quad \text{with}: \quad \kappa = \frac{c_{it}}{c_{iv}}. \tag{16}$$

While the cost of each integration and verification might be thought of as the monetary expense of assembling a complete product based on a number of computer-aided design files or performing a system analysis, the cost of each (local) iteration can be seen as the financial expense incurred by each stakeholder individually when a component design is modified and evaluated with respect to the given requirements without informing others about it.

We study the tradeoff between task completion time and theoretical cost based on two illustrations: Figure 9a, which shows all the task completion times and associated costs for $\kappa = 0.1$, meaning that each assembly costs 10 times as much as each local design change – a reasonable assumption when compared to reality. Figure 9b illustrates the centre of each data cluster, that is, the median of all task completion times versus the median of all theoretical costs, when the costs are computed for three different scenarios ($\kappa = 0.1, \kappa = 1, \kappa = 10$). Each scenario is additionally approximated by a first-order polynomial regression. The subjects were not informed about any costs or cost analysis performed afterwards.

Description of results. In Figure 9a, the samples for each time interval form separate clusters at different locations inside the parameter space spanned by the task completion time and theoretical process costs. While design tasks with little time between each integration and verification are characterised by relatively high costs and low task completion times, those with long time intervals between each integration and verification display relatively low costs and high task completion times. Besides the location of the clusters, their shape reveals another important finding. In case of 0 seconds time intervals, for example, the data is spread over a rather broad area, whereas for intervals of 3, 6, 9 and 12 seconds, the data points approximate a straight line.

The centres of gravity of the clusters (pairs of median values) shown in Figure 9a are depicted in Figure 9b (shown as plus signs). Both their location and the first-order polynomial regression (for $\kappa = 0.1$) confirm the previous observations. The slope of the regression model depicts the tradeoff between development time and cost for a varying integration and verification frequency. Its negative incline shows that both performance measures are subject to a conflict of goals (each one can only be improved at the expense of the other).

With an increasing cost ratio κ, the regression model and the tradeoff between development time and cost change significantly. A tipping point can be observed at about $\kappa = 1$. In this case (where each integration and verification costs as much as each local design iteration), development time can be reduced at constant cost if the integration and verification frequency is increased. For even higher values of κ, the regression model has a positive slope. This means that both development time and process costs can be reduced simultaneously if the time interval between integration and verification is shortened. Thus, in this case, there exists no conflict of goals.

https://doi.org/10.1017/dsj.2022.26 Published online by Cambridge University Press

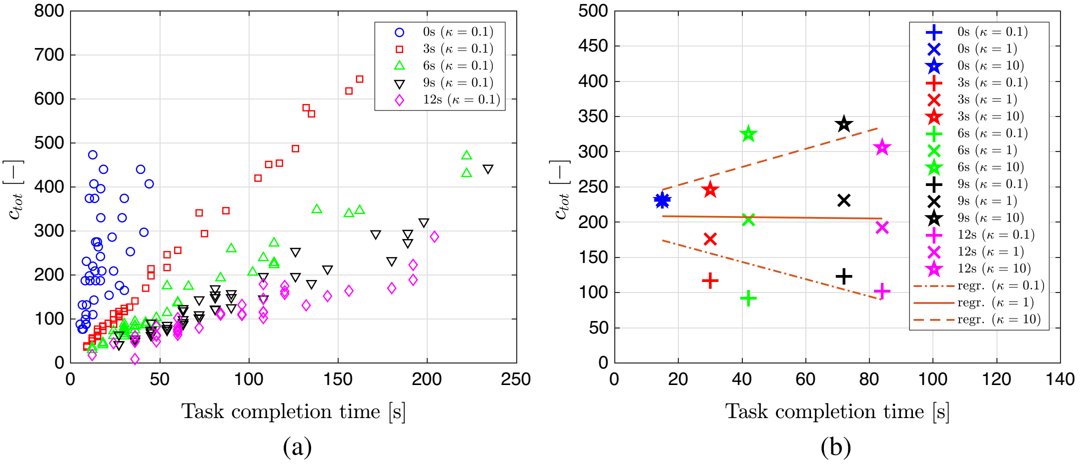

Figure 9. Influence of varying time interval (between each integration and verification) on task completion time and theoretical cost for: (a) a fixed ratio of $\kappa = 0.1$ and (b) median values and first-order polynomial regression for a varying κ.

6. Discussion

In this section, we discuss the methodical research approach and the experimental results. At first, we interpret the outcome of the single- and multi-actor study and suggest possible causes for the observed effects. Then, we present the limitations of our work. And finally, we outline the implications for academia and industry.

6.1. Interpretation of the results

The following explanations for the results of the single- and multi-actor study are based on logical reasoning and the analysis of a survey which subjects responded to after each session. Direct quotes are translated from German into English.

Single-actor study

Based on the high level of agreement between our experimental data and the data from Grogan & de Weck (2016), two major findings can be presented: first, the impact of task size on task completion time for design tasks with two, three and four input and output variables, for which Grogan & de Weck (2016) provide a statistical correlation, is confirmed. In this sense, our work might be considered a replication study that verifies some of the earlier findings in this field. Second, our experimental setup (GUI, back-end and subject selection) is found to be reliable and produces similar results as previous studies conducted under the same boundary conditions.

The minor deviations between our data and the data from Grogan & de Weck (2016) might be attributed to the slightly different GUI (e.g., buttons above and below the input sliders), the demographics of the subjects (e.g., work experience, average age) and the network delays due to the virtual format of the study.

https://doi.org/10.1017/dsj.2022.26 Published online by Cambridge University Press

Multi-actor study

The outcome of the multi-actor study provides numerous insights. First of all, we are able to confirm that varying the frequency of integration and verification has a significant impact on the task completion time. On average, shorter time intervals between each integration and verification allow subjects to solve the design tasks faster (40–71% reduction of task completion time if the time interval is cut in half). This effect could be due to the rapid information exchange, as any update of an input variable is instantly mapped onto all other output variables and only little time is 'lost' by waiting for design changes that subjects require to reach the own target area. For low integration and verification frequencies, one subject, for example, noted that: '*I frequently had to wait for a long time until I could see the effect of my partner's input variable*'. Fast feedback on design changes is found to be an important factor influencing the task completion time as another subject added: '*Late feedback made it harder to find a solution*'.

Some also tried to anticipate the decision-making of others while observing the impact of their actions on the own output variables. Here, frequent integration and verification also made it easier for subjects to understand the actions of their counterpart. One subject, for example, mentioned: '*With a time delay, it is more difficult to grasp the behaviour of the other person*'.

Frequent integration and verification, however, can, in some cases, also lead to long task completion times, as evidenced by outliers and uncompleted tasks. This might be because some subjects are irritated and stressed by the rapid and at times unpredictable movements of their output slider as the other subject randomly tests different locations of their input slider. One subject stated: '*With a time delay, it was significantly more comfortable as it made the collaboration calmer, with no hectic back and forth movements of the sliders – way more transparent*'. Another subject said: '*Without any time delay, it almost felt like we were fighting against each other*'. Hence, high integration and verification frequencies can also induce difficulties for subjects, which may even lead to higher task completion times. We expect this effect to be even stronger in case of larger (and more complex) design tasks that represent real-world development problems more accurately. Feedback from designers of an automotive company revealed that design teams sometimes even refuse to incorporate new design information provided to them so that they can optimise their subsystem without disturbance.

Besides the integration and verification frequency, other factors also influence the task completion time. Our results suggest two additional factors of influence: first, the composition of the test groups, that is, the allocation of subjects into pairs, which always combines two specific decision-making strategies. If these provoke some kind of self-excited oscillations, it might favour high task completion times. One of our test groups, for instance, is responsible for 2 of the 14 outliers and 3 of the 4 uncompleted tasks, which corresponds to 28% of the 18 design tasks that are either outliers or uncompleted. This effect, however, has not been statistically examined. The second factor is the coupling strength of a system, which appears to have a considerable influence on the task completion time. We are able to show that low trace values favour high task completion times, that is, it takes considerably longer to solve a design task when subjects influence the output variables of other subjects more than their own one. This seems reasonable, as subjects who have a greater effect on others than on themselves can create considerable disturbances when moving their own input slider and, thus, induce additional design iterations.

https://doi.org/10.1017/dsj.2022.26 Published online by Cambridge University Press

An exception is the time interval of 0 seconds between each integration and verification. In this specific case, the task completion time neither increases nor decreases when the trace is varied (if the trace is generally high). Hence, a frequent information flow appears to stabilise the task completion time even when the coupling strength of a system rises or falls. This might be because subjects receive immediate feedback on their design decisions and notice that their behaviour is causing difficulties for others. However, no subject mentioned any experiences related to this.

Our results also highlight the relationship between task completion time and theoretical cost when varying the integration and verification frequency. We show that the tradeoff between both metrics is greatly dependent on the assumed cost of each local design iteration and each integration and verification. In particular, a conflict of goals between short task completions times and low cost emerges, if each integration and verification costs significantly more than each local iteration. This is because, in this scenario, the overall costs are dominated by the integration and verification steps, which occur frequently and induce high specific costs each time. Each additional integration and verification therefore significantly increases the accumulated cost, while also allowing a reduction in the task completion time. If each local design iteration costs as much as (or more than) each integration and verification, this conflict of goals resolves as each integration and verification step has less influence on the overall cost. Thus, in such situations, more frequent integration and verification results in shorter task completion times and constant (or even lower) costs.

6.2. Limitations and constraints

Unfortunately, our research approach is also subject to certain limitations. One is the parameter design framework itself, which greatly simplifies design processes, as it neglects several factors of influence, such as domain-specific know-how and teamwork. In addition, subjects are uninformed about the technical dependencies between their input and output variables. In reality, however, designers often use analytical formulas and surrogate models in order to comprehend the behaviour of their system, which may enable them to more quickly identify a proper design. This might also be supported by direct communication (talking, messaging, etc.) between subjects, which we did not permit.

Another limiting factor is the size and structure of the surrogate design tasks used in both studies. Even though small-scale application problems, like the 2×2, 3×3 and 4×4 design tasks, allow transparency and traceability, they only represent reality to a certain degree, as real-world design problems contain more design variables and quantities of interest. Usually, they also include product properties on multiple hierarchical levels instead of just two, for example, $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ where $\mathbf{y} = f(\mathbf{x})$, $\mathbf{z} = f(\mathbf{y})$. This means that, in terms of their responsibility, subjects are positioned side by side as well as above and below each other. Finally, note that the physical dependencies between product properties are not always linear (as assumed in our case) but can also be nonlinear, which might make design tasks more difficult to solve. Therefore, it is difficult to assess the degree to which our findings can be generalised to larger and more complex design tasks, which might more precisely reflect actual development processes.

https://doi.org/10.1017/dsj.2022.26 Published online by Cambridge University Press

In reality, complex systems also require integration and verification processes that are more advanced than the ones for which we provide a theoretical approach. Often times, for example, subsystems are assembled asynchronously rather than synchronously, which means that some parts of the complete product are merged more frequently than others, for example, because many design changes occur or because each integration and verification comes at a low cost. The framework we suggest, however, only considers synchronous product integration. In real-world product design scenarios, integration and verification phases also have a specific duration, that is, $t_{int} \neq 0$ and $t_{ver} \neq 0$. For the sake of simplicity, we did not take this into consideration.

A final limitation concerns the generic cost model, which we use to study the tradeoff between the task completion time and the process costs. This is a strong simplification of the actual costs that arise during real-world product development processes, in particular, because it combines many of the typical cost factors and assumes a linear relation between the independent variables and the overall cost.

6.3. Implications and relevance

Our work contributes to the field of engineering design by applying an established framework (parameter design) to a new research problem (influence of integration and verification frequency on task completion time and costs). In the following, we differentiate between the implications of this study for academia and industry.

Academia

In terms of its scientific contribution, our work may provide progress in two ways: first, it enhances the understanding of coordination processes like integration and verification in distributed design, since we explore the benefits and limitations of different process configurations quantitatively. In this respect, our insights along with findings of future studies based on our theoretical approach could be useful. And second, our experimental data provides a basis for analytical process models to be calibrated with. This could be significant as many analytical process models lack empirical validation due to the absence of objective data.

Industry

In terms of its practical contribution, our study might be beneficial to those who manage integration and verification processes in large-scale companies. Here, we provide three helpful guidelines (assuming that integration and verification itself takes no time): first, if the only goal is to reduce development time, the integration and verification frequency should be as high as possible. Second, if process costs are also considered and each integration and verification costs much more than each local design iteration, the integration and verification frequency is typically subject to a conflict of goals between reaching a low development time and a low overall cost. This means that the integration and verification frequency should be chosen carefully based on the desired development time and the available budget. And third, if each integration and verification costs as much as or much less than each local design iteration, higher integration and verification frequencies should yield lower development times at constant or even lower costs. Thus, in this case, the shortest interval between each integration and verification should be chosen.

https://doi.org/10.1017/dsj.2022.26 Published online by Cambridge University Press

The dynamic tradeoff between development time and cost might be significant for organisations that are about to implement digital methods in their engineering design processes, in particular because those methods usually reduce the specific cost of each integration and verification (causing a change in κ from low to high). Thus, the adoption of digital and model-based methods may (at some point) lead to a tipping point at which higher integration and verification frequencies are better not only regarding development time but also regarding cost.

7. Conclusion

In this paper, we presented an empirical study on the effect of different integration and verification frequencies in distributed design processes. For this purpose, we extended the established parameter design framework of Hirschi & Frey (2002) and Grogan & de Weck (2016) to account for three distinct phases that subjects pass through iteratively as they take part in a multi-actor experiment. Those three phases are isolated design, integration and verification.

A single-actor study to validate our methodical approach was conducted with 34 subjects who solved 291 coupled parameter design tasks with 2, 3 and 4 input and output variables. This study confirmed previous findings in this field (Grogan & de Weck 2016) as well as the reliability of our setup. A second (multi-actor) study required the same subjects (in groups of 2) to solve 229 coupled parameter design tasks with 2 input and 2 output variables, whereby the time between each integration and verification was varied between 0, 3, 6, 9 and 12 seconds. This led to the following three insights: first, on average, higher integration and verification frequencies allowed subjects to solve design tasks significantly faster. Doubling the time interval between each integration and verification, for example, resulted in an increase in task completion time of about 40–71%. Second, the coupling strength of a system, which in our study also marked the extent to which a subject affected his or her own output variable compared to the overall coupling strength, also had a considerable influence on the task completion time. At high integration and verification frequencies, however, this effect seemed to disappear. And third, when taking into consideration potential process cost, a conflict of goals emerged between attaining a short task completion time and a low cost if the specific cost of each integration and verification step was significantly higher than that of each local design iteration. However, if the specific cost of each local design iteration was significantly higher than that of each integration and verification step, we did not detect any conflict of goals and higher integration and verification frequencies resulted in both shorter task completion times and lower overall costs.

Note that considerable limitations exist which might affect the generalisability and validity of our results.

8. Outlook

Future work in this field could have three goals: first, to analyse our experimental data in more detail, that is, on the level of individual design decisions represented by the movements of the input sliders. This would allow more conclusions regarding the design behaviour of the individual subjects. Second, to repeat our study using different boundary conditions (e.g., larger design tasks, nonlinear dependencies, asynchronous integration and verification phases). Third, to extend

https://doi.org/10.1017/dsj.2022.26 Published online by Cambridge University Press

the parameter design approach even further to enable the investigation of design tasks with three or more hierarchical levels in which subjects are positioned side by side as well as above and below each other in terms of their assigned responsibility.

Acknowledgments

We wish to thank Mr. Paul T. Grogan for providing the open-source software that allowed us to perform this study. We also would like to thank all the subjects who took part in this study.

Financial support

This research was funded by BMW.

References

Alelyani, T., Yang, Y. & Grogan, P. T. 2017 Understanding designers behavior in parameter design activities. In *ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, V007T06A030, August 6–9, 2017, Cleveland, Ohio, USA*; doi: 10.1115/DETC2017-68335.

Austin-Breneman, J., Honda, T. & Yang, M. C. 2012 A study of student design team behaviors in complex system design. *Journal of Mechanical Design* **134** (12), 124504; doi: 10.1115/1.4007840.

Borji, A. & Itti, L. 2013 Bayesian optimization explains human active search. In *Advances in Neural Information Processing Systems 26*. (**ed.** C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani & K. Q. Weinberger), pp. 55–63. Curran Associates.

Chatfeld, M. & Mander, A. 2009 The Skillings–Mack test (Friedman test when there are missing data). *The Stata Journal* **9**, 299–305.

Chaudhari, A. M. & Panchal, J. H. 2019 An experimental study of human decisions in sequential information acquisition in design: impact of cost and task complexity. In *Research into Design for a Connected World (vol. 134)*, pp. 321–332. Springer.

Clarkson, P. J. & Hamilton, J. R. 2000. 'Signposting': a parameter-driven task-based model of the design process. *Research in Engineering Design* **12**, 18–38.

Egan, P., Schunn, C., Cagan, J. & LeDuc, P. 2015 Improving human understanding and design of complex multi-level systems with animation and parametric relationship supports. *Design Science* **1**, e3; doi:10.1017/dsj.2015.3.

Flager, F., Gerber, D. J. & Kallman, B. 2014 Measuring the impact of scale and coupling on solution quality for building design problems. *Design Studies* **35** (2), 180–199; doi: 10.1016/j.destud.2013.11.001.

Goodman, T. & Spence, R. 1978 The effect of system response time on interactive computer aided problem solving. *ACM SIGGRAPH Computer Graphics* **12** (3), 100–104; doi: 10.1145/965139.807378.

Grogan, P. T. 2019 *Collaborative Design Web Platform (collab-web)*. GitHub. Retrieved from https://github.com/code-lab-org/collab-web.

Grogan, P. T. & de Weck, O. L. 2016 Collaboration and complexity: an experiment on the effect of multi-actor coupled design. *Research in Engineering Design* **27**, 221–235; doi: 10.1007/S00163-016-0214-7.

https://doi.org/10.1017/dsj.2022.26 Published online by Cambridge University Press

Haskins, C., Forsberg, K., Krueger, M., Walden, D. & Hamelin, D. 2006 *Systems Engineering Handbook*. INCOSE.

Hirschi, N. 2000 *Design, cognition and complexity: an investigation using a computer based design task surrogate* (Master Thesis), MIT.

Hirschi, N. & Frey, D. 2002 Cognition and complexity: an experiment on the effect of coupling in parameter design. *Research in Engineering Design* **13**, 123–131; doi:10.1007/s00163-002-0011-3.

Lewis, K. & Mistree, F. 1997 Modeling interactions in multidisciplinary design: a game theoretic approach. *AIAA Journal* **35** (8), 1387–1392.

Lewis, K. & Mistree, F. 1998 Collaborative, sequential and isolated decisions in design. *ASME Journal of Mechanical Engineering* **120** (4), 643–652.

Lindemann, U., Maurer, M. & Braun, T. 2008 *Structural Complexity Management: An Approach for the Field of Product Design*. Springer.

Mann, H. B. & Whitney, D. R. 1947 On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics* **18** (1), 50–60.

McComb, C., Cagan, J. & Kotovsky, K. 2015 Rolling with the punches: an examination of team performance in a design task subject to drastic changes. *Design Studies* **36**, 99–121; doi:10.1016/j.destud.2014.10.001.

Mihm, J., Loch, C. & Huchzermeier, A. 2003 Problem-solving oscillations in complex projects. *Management Science* **49** (6), 733–750.

Panchal, J. H., Sha, Z. & Kanna, K. N. 2017 Understanding design decisions under competition using games with information acquisition and a behavioral experiment. *Journal of Mechanical Design* **139** (9), 091402; doi:10.1115/1.4037253.

Panchal, J. H. & Szajnfarber, Z. 2020 Experiments in systems engineering and design research. *Systems Engineering* **20** (6), 529–541; doi:10.1002/sys.21415.

Pimmler, T. U. & Eppinger, S. D. 1994 Integration analysis of product decompositions. In *ASME 1994 Design Technical Conferences collocated with the ASME 1994 International Computers in Engineering Conference and Exhibition and the ASME 1994 8th Annual Database Symposium*, pp. 343–351. ASME; doi:10.1115/DETC1994-0034.

Rötzer, S., Schweigert-Recksiek, S., Thoma, D. & Zimmermann, M. 2022 *Attribute dependency graphs: modelling cause and effect in systems design, Design Science* Vol. 8, e27, DOI: https://doi.org/10.1017/dsj.2022.20.

Royston, P. 1992 Approximating the Shapiro–Wilk W-test for non-normality. *Statistics and Computing* **2**, 117–119.

Sha, Z., Kannan, K. N. & Panchal, J. H. 2014 Behavioral experimentation and game theory in engineering systems design. *Journal of Mechanical Design* **137** (5), 051405; doi: 10.1115/1.4029767.

Shapiro, S. S. & Wilk, M. B. 1965 An analysis of variance test for normality (complete samples). *Biometrika* **52** (3/4), 591–611.

Simpson T. W., Barron, K., Rothrock, L., Frecker, M., Barton, R. R. & Ligetti, C. 2007 Impact of response delay and training on user performance with text-based and graphical user interfaces for engineering design. *Research in Engineering Design* **18**, 49–65; doi:10.1007/s00163-007-0033-y.

Simpson, T. W., Iyer, P., Rothrock, L., Frecker, M., Barton, R., Barron, K. & Meckesheimer, M. 2005 Metamodel-driven interfaces for engineering design: impact of delay and problem size on user performance. In *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*. AIAA; doi:10.2514/6.2005-2060.

Skillings, J. H. & Mack, G. A. 1981 On the use of a Friedman-type statistic in balanced and unbalanced block designs. *Technometrics* **23** (2), 171–177.

https://doi.org/10.1017/dsj.2022.26 Published online by Cambridge University Press

Stanglmeier, M. 2018 *Methode zur monetären Bewertung virtueller Absicherungsprozesse* (PhD Thesis), Otto-von-Guericke-Universität Magdeburg.

Stanglmeier, M., **Schäfer, C.**, **Wandt, R.** & **Schenk, M.** 2018 Economic assessment of virtual validation processes in the automotive bdevelopment: an analytical approach. In *2018 Annual IEEE International Systems Conference (SysCon)*. IEEE; doi:10.1109/SYSCON.2018.8369496.

Summers, J. D. & **Shah, J. J.** 2010 Mechanical engineering design complexity metrics: size, coupling, and solvability. *Journal of Mechanical Design* **132** (2), 021004; doi: 10.1115/1.4000759.

Szajnfarber, Z., **Grogan, P. T.**, **Panchal, J. H.** & **Gralla, E.** 2020 A call for consensus on the use of representative model worlds in systems engineering and design. *Systems Engineering* **23**, 436–442; doi:10.1002/sys.21536.

Ulrich, K. & **Eppinger, S. D.** 2015 *Product Design and Development* (sixth edition). McGraw Hill/Irwin.

VDI 2206 2004 *Entwicklungsmethodik für mechatronische Systeme—Design Methodology for Mechatronic Systems*. Beuth.

Wöhr, F., **Königs, S.**, **Ring, P.** & **Zimmermann, M.** 2020 A role-activity-product model to simulate distributed design processes. In *Proceedings of the 22nd International DSM Conference*. MIT.

Wöhr, F., **Stanglmeier, M.**, **Königs, S.** & **Zimmermann, M.** 2020 Simulation of gradient-based individual design behaviour in distributed development processes. In *Proceedings of the Design Society: DESIGN Conference* (vol. **1**), pp. 1579–1588. Cambridge University Press; doi:10.1017/dsd.2020.51.

Wynn, D. C., **Eckert, C. M.** & **Clarkson, P. J.** 2006 Applied signposting: a modeling framework to support design process improvement. Presented at the ASME International Design Engineering Technical Conferences, Philadelphia, PA, 2006.

Yassine, A., **Joglekar, N.**, **Braha, D.**, **Eppinger, S.** & **Whitney, D.** 2003 Information hiding in product development: the design churn effect. *Research in Engineering Design* **14**, 145–161.

Yu, B. Y., **de Weck, O.** & **Yang, M. C.** 2016 Parameter design strategies: a comparison between human designers and the simulated annealing algorithm. In *ASME 2015 (vol. 7).* The American Society of Mechanical Engineers; doi:10.1115/DETC2015-47674.

Yu, B. Y., **Honda, T.**, **Sharqawy, M.** & **Yang, M.** 2016 Human behavior and domain knowledge in parameter design of complex systems. *Design Studies* **45** (Part B), 242–267; doi:10.1016/j.destud.2016.04.005.

Zimmermann, M., **Königs, S.**, **Niemeyer, C.**, **Fender, J.**, **Zeherbauer, C.**, **Vitale, R.** & **Wahle, M.** 2017 On the design of large systems subject to uncertainty. *Journal of Engineering Design* **28** (4), 233–254; doi:10.1080/09544828.2017.1303664.

https://doi.org/10.1017/dsj.2022.26 Published online by Cambridge University Press