DPCM: A method for modelling and analysing design process changes based on the Applied Signposting Model

Daniel Shapiro*, Matthew D. Curren** and P. John Clarkson*

Engineering Design Centre, University of Cambridge, Cambridge, UK

*** Fan & Compressor Sub-System, Civil Large Engines, Rolls-Royce PLC, Derby, UK*

Please contact the corresponding author, Daniel Shapiro, by e-mail: ds678@cam.ac.uk.

DPCM: A method for modelling and analysing design process changes based on the Applied Signposting Model

Research on changes in design has focused on changes in the product domain. However, because the product's design process may change as well, this article suggests a comprehensive method to support modelling and analysing changes in the process domain (DPCs). After developing the concept for the Design Process Change Method (DPCM) based on requirements derived from literature and industrial practice, the DPCM is detailed and computationally implemented using the framework of the Applied Signposting Model. The DPCM enables design teams to conduct various useful analyses, which enhance the understanding of DPC effects on process performance, support process execution through suggesting reactions to DPCs, and support process planning through identifying and prioritising the 'right' DPCs. The method's application is demonstrated based on the fan sub-system preliminary design process of Rolls-Royce PLC.

Keywords: process modelling; process change; iteration; process simulation; Design Process Change Method (DPCM)

1. Introduction

Product development (PD) is a key function in industrial organisations and crucial for their commercial success. Fierce competition has put pressure on companies to develop cheaper products of higher quality in less time and to fulfil rapidly changing customer needs. Also, decreasing technology life cycles and an increasing technological diversity have amplified the pace and complexity of PD. This has drawn much attention to the management of design processes, which encompass a spectrum of activities at the core of PD and aim at creating recipes for the production of products.

Both the dynamic and complex environment of PD as well as the inherently uncertain nature of innovative design processes lead to an industrial reality in which engineering changes (EC), loosely defined as changes in released engineering documentation, are very common. Consequently, since the late 90's many tools for engineering change management (ECM) have been developed (see Hamraz et al. 2013a for examples). However, not only are ECs likely to occur, i.e. changes in the product domain, but also changes in the process domain, for example delays in activities, unplanned iterations or the addition of new activities to the process plan. In fact, whenever an EC arises, the process plan may need to be amended since inputs of activities change (Chua and Hossain 2012). Because such design process changes (DPCs) can propagate leading to rework and indirectly affecting numerous activities and deliverables in the process, it can be particularly difficult to predict their overall impacts on key process performance metrics like process duration and development cost (Shapiro, Sommer, and Clarkson 2015). However, the impacts of DPCs can be considerable: In a study of 448 technological projects, Dvir and Lechler (2004) found that the only distinguishing factor between successful and failed technological projects, independent of their innovativeness, was the amount of goal and plan changes during project execution. Karniel and Reich (2013, p. 208) also acknowledged the relevance of DPCs and observed that so far "the typical practice has been reactively following changes… rather than proactively planning through analysis of potential changes."

The numerous existing activity-network based modelling tools that support the management of design processes usually assume that sufficient knowledge exists a-priori to plan the design process and execute it accordingly. However, this assumption often proves inadequate leading Karniel and Reich (2013) to the conclusion that managerial issues associated with DPCs are insufficiently addressed by existing methods. This is also supported by a prior study of the authors, who found only 27 existing methods that account for DPCs. Since all these methods comprise different features and offer varying degrees of support the authors recognised the need for the systematic development of a new comprehensive support method, which helps design teams account for the impacts of DPCs on design process performance during process planning and execution (Shapiro, Sommer, and Clarkson 2015).

This article describes the systematic development of such a support method, which is called the Design Process Change Method (DPCM) hereafter. This method will be based on the Applied Signposting Model (ASM; Wynn, Eckert, and Clarkson 2006), an activity-network-based framework for design process modelling. The rest of this article is organised as follows: Section 2 provides an overview on DPCs and existing support methods; Section 3 explains the research method; Section 4 examines the conceptual design of the DPCM; Section 5 elaborates on the method's detail design; Section 6 describes the application of the DPCM to the fan sub-system preliminary design process at Rolls-Royce PLC; Section 7 discusses the method's practical usefulness as well as the key assumptions made and directions for future research; Section 8 summarises and concludes the article.

2. Design Process Changes

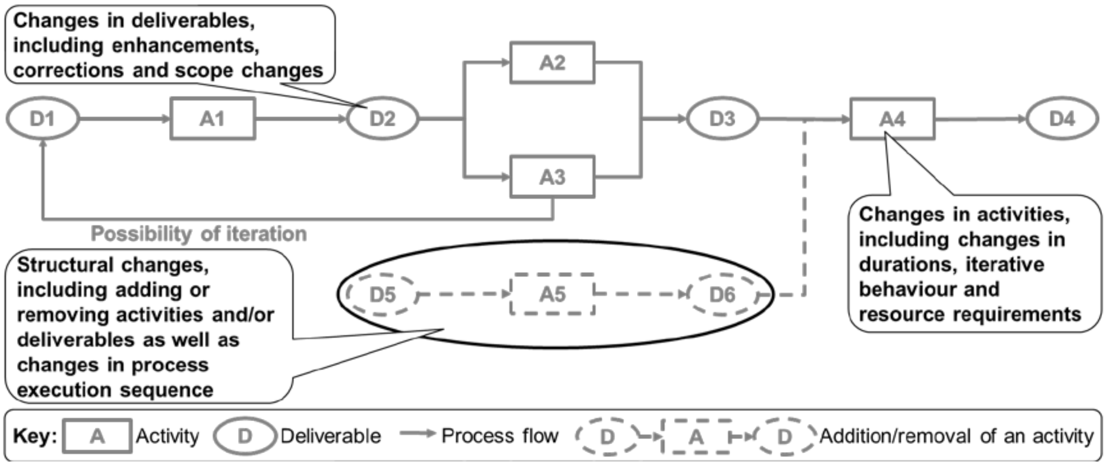

Figure 1. Major DPC types shown in an idealised design process model. Adapted from Shapiro, Sommer, and Clarkson 2015.

This section summarises major DPC characteristics, including their reasons, types, and consequences, as well as features of existing support methods. It is based on a recent literature survey by the authors, who define DPCs as "changes and/or modifications… to planned design activities (involved resources, tools, etc.), their resultant deliverables (drawings, documents, prototypes and generally descriptions of the technical artefact) or the relationships between design activities and deliverables (process structure)" (Shapiro, Sommer, and Clarkson 2015).

This definition emphasises three fundamental types of DPCs (see Figure 1), i.e. changes in activities, deliverables and structural changes, and is also consistent with most activity network-based design process models, which view processes as discrete activities interconnected through deliverables. Changes in activities refer to changes in their attributes, including durations, their iterative behaviour and resource requirements. Changes in deliverables comprise enhancements, corrections and scope changes in already produced activity inputs or outputs. As such they are similar to ECs which denote changes of released product descriptions, but only presume that a product description has been created and not necessarily released. Lastly, structural changes describe all DPCs that affect the process scheme, including adding or removing activities or deliverables as well as changing the process execution sequence, i.e. activity order or degree of concurrency. Structural changes immediately affect at least one activity and one deliverable in the process. If, for example, a new activity is added to the process, it might require existent deliverables as inputs and will necessarily produce a new output or change an existing output.

The existing DPC literature mostly discusses DPCs that originate from ECs (Chua and Hossain 2012; Li and Moon 2012; Wynn et al. 2014) and thus, are triggered by product-related reasons. When ECs occur, deliverables, which are inputs (and outputs) of design activities, need to be altered and lead to rework of the respective activities. However, as the design process involves the cooperation of people across multiple organisations there are many other reasons which do not necessarily originate from the product. For example, DPCs may come from process improvements suggested by project-team members or new managers, or originate from designers' lack of competence or too optimistic plans. Another major reason for DPCs is a shortage in available resources during process execution, including manpower, information, facilities and funding (Dvir and Lechler 2004). Lastly, DPCs can be caused by other DPCs as they can propagate throughout the design process. For example, an activity requiring rework because of a change in its input might result in a changed output, which serves as an input to subsequent activities that consequently might also require rework, and so on (Ouertani 2008).

The various DPC types can affect process duration (Karniel and Reich 2013), effort (Cronemyr, Öhrwall Rönnbäck, and Eppinger 2001) and potentially even product quality (Li and Moon 2012), although only very few publications mention effects on the latter. A major effect of all DPC types is that they can cause iteration (Chua and Hossain 2012). This and propagation make it difficult to predict DPC effects on process performance.

As the few existing methods for modelling and analysing DPCs are often inspired by EC-propagation methods in the product domain, they have analogous goals (Hamraz, Caldwell, and Clarkson 2013; Shapiro, Sommer, and Clarkson 2015), i.e. to support design teams to

(1) Gain understanding of DPC effects on design process performance;
(2) Improve process execution by reacting to and implementing DPCs efficiently;
(3) Improve process planning by prioritising optional DPCs effectively based on costs/benefits.

However, compared to the rich literature on ECM there is a lack of comprehensive methods to support management of DPCs (Karniel and Reich 2013). In fact, the authors' literature survey (Shapiro, Sommer, and Clarkson 2015) identified only 27 methods among which 18 solely examine effects of deliverable changes (e.g., Wynn, Caldwell, and Clarkson 2014; Ouertani 2008). Among the other nine methods there are none which cover all three major DPC types. Also, the reviewed methods offer highly varying features: Some methods, for example, can help identifying activities affected by a DPC (Ahmad, Wynn, and Clarkson 2013), while other methods additionally suggest an activity sequence for DPC implementation (Khoo, Chen, and Jiao 2003). Furthermore, other methods consider the implementation of specific DPCs (Chua and Hossain 2012), while some assume stochastic DPC arrival rates (Li and Moon 2012).

One common feature among most methods (although treated differently) is that some sort of change propagation is represented.

Overall, the many possible reasons for DPCs, their potentially severe impacts and the lack of a comprehensive support indicate the need for a novel method for modelling and analysing DPCs (Shapiro, Sommer, and Clarkson 2015).

3. Research method

To address the need of developing a comprehensive method for modelling and analysing DPCs, as identified in Section 2, this research follows a systematic procedure analogously to actual engineering design processes (Cross and Roozenburg 1992), i.e. deriving requirements, developing alternative concepts, selecting and elaborating a concept, detailing and implementing as well as evaluating and refining the method. While the method's conceptual design is described in Shapiro and Clarkson 2016, this article focusses primarily on the method’s detail design and evaluation. Nevertheless, a comprehensive summary of the method’s conceptual design is provided in Section 4.

The research thus, started with a requirements analysis to define the specific needs that the DPCM should fulfil. Subsequently, a morphological chart was developed to convert the derived functional requirements into alternative concept ideas. A selection among these concept ideas was then made to form a broad overall concept for the DPCM. The authors then described the method’s fundamental elements and their interrelations and defined a set of analyses to improve process understanding, planning and execution, which will be enabled through such a support method. The rationale for choices in the method’s conceptual design (e.g., requirements, alternative concept ideas, a single broad concept etc.) was based on three sources: first, the authors' literature survey on DPCs (Shapiro, Sommer, and Clarkson 2015) and other key engineering design literature; second, the authors' literature-based exploratory study of a high-speed machining device's design process (Shapiro et al. 2015); third, the study of the fan sub-system preliminary design process at Rolls-Royce PLC (see Section 6).

The authors then converted the conceptual descriptions of method elements and their relationships into detailed definitions and quantifiable functions, which can be specified in practice and implemented computationally. Also the detailed implementation of the previously discussed analysis set was specified. Finally, the developed DPCM was applied to the fan sub-system preliminary design process, which was studied over a time period of seven months: At the beginning, six semi-structured interviews were conducted with three design engineers and their manager to review an existing process flowchart and gain understanding of the design process. Then, the first data collection workshop was carried out, which was composed of hourly time slots with each of five design engineers and the manager with the aim of collecting fundamental data about activity durations (probability distributions), resource requirements and iteration-likelihoods to build an ASM model of the design process. The model was used to run basic process simulations that produced frequency distributions of the overall process duration and effort, the latter being measured in person-days. These results were then reported back to the design team, who verified their plausibility based on the team’s experience with numerous past executions of this design process for different jet engine models. Overall, the accuracy of the data collected in the first workshop could be thus, positively confirmed. Subsequently, the second workshop was held over half a day with two design engineers to collect additional data for the application of the DPCM, for example, effort level boundaries, confidence mappings and multiplicative factors for iteration-likelihoods (see Section 5 for explanations of these variables). Afterwards, the DPCM’s analyses could be conducted and produced advanced results as reported in Section 6. Finally, these advanced results were discussed in two meetings with the design team, who confirmed their plausibility and practical usefulness (see Section 7.1).

4. Conceptual design of the DPCM

This section describes the DPCM's conceptual design, including the method's fundamental elements, relationships and potential applications. Although, the major procedure and assumptions underlying the conceptual design as well as its results are discussed in the following, the interested reader is also referred to Shapiro and Clarkson 2016, which examines the method's conceptual design in even greater detail.

To begin with, Table 1 summarises the identified functional requirements for a comprehensive DPC support method, some selected sources per requirement and broad conceptual ideas how the DPCM is intended to fulfil these. As no literature was found covering an explicit discussion of requirements for such a method, the functional requirements were derived based on the features of the methods described in Section 2 and based on an exploratory case study (Shapiro et al. 2015). Moreover, some general requirements for method inputs, application and outputs were taken from the very comprehensive, requirements-based development of an ECM tool by Hamraz et al. (2013) and transferred from the product- into the process-domain. This approach thus resulted in a list of functional and general method requirements (please contact the first author for the full list), of which only the functional requirements are presented in Table 1 due to their predominant importance for the conceptual design of the DPCM.

Table 1: Conceptual ideas to fulfil the identified functional method requirements.

Functional requirement	Conceptual idea to fulfil the requirement
1. Activity-based modelling of evolutionary design processes (Khoo, Chen, and Jiao 2003; Wynn, Caldwell, and Clarkson 2014)	Theoretically, the DPCM could be based on various activity-network-based frameworks. The ASM (Wynn, Eckert, and Clarkson 2006) is suggested because it allows capturing complex interrelations between activities and deliverables, which is key for the analysis of DPCs. Also, it was specifically developed for modelling design processes and thus, contains many design-focused features so that it is convenient to use.
2. Modelling iteration (Chalupnik et al. 2007; Li and Moon 2012)	As DPCs can affect the level of uncertainty in the process and may trigger iterations, which substantially impact process performance (Eppinger 1991), it is suggested to model the occurrence of iterations dependent on uncertainty (see, e.g., Lévárdy and Browning 2009).
3. Modelling changes in activities (Cronemyr, Öhrwall Rönnbäck, and Eppinger 2001; Khoo, Chen, and Jiao 2003)	It is suggested to represent changes in activities as changes in the associated effort (Lukas et al. 2007), which implicitly considers both changes in activity durations and in their resource requirements, and increases the flexibility of modelling real-world processes.
4. Modelling changes in deliverables (Chua and Hossain 2012; Wynn, Caldwell, and Clarkson 2014)	Based on the existing literature (see, e.g., Chua and Hossain 2012), it is suggested to account for specific changes in deliverables by capturing their potential of causing iterations.
5. Modelling structural changes (Karniel and Reich 2013)	It is suggested to adapt process plans manually in order to represent structural changes because rule-based automatic adaptation of plans adds significant complexity to the method and also does not work for every change case (Karniel and Reich 2013). The DPCM's user will thus decide, e.g., whether a new activity can be added without adding new deliverables.
6. Modelling propagating DPC effects (Ahmad, Wynn, and Clarkson 2013; Ouertani 2008)	It is suggested to consider DPC propagation between activities and deliverables (Wynn, Caldwell, and Clarkson 2014) in order to equally capture the lower-level effects of changes in activities, deliverables and structural changes, i.e. activity-deliverable relationships. Moreover, in order to limit the complexity of modelling, the method concept is restricted to the process domain so that change propagation to elements of the product domain, such as components, is not considered.

7. Identifying critical DPCs, reactions and DPC-based process improvements (Chalupnik et al. 2007; Cronemyr, Öhrwall Rönnbäck, and Eppinger 2001)	To increase the practical usefulness the identification and comparison of multiple alternative candidates for critical DPCs, reactions and process improvements (Browning and Eppinger 2002), rather than the identification of a single theoretical worst or best case, should be supported.
8. Analysing DPC impacts (Chua and Hossain 2012; Lukas et al. 2007)	Process simulations are suggested to assess DPC impacts because closed-form analysis is often not possible for complex, stochastic networks (Shapiro et al. 2015).

The method's concept integrates the described conceptual ideas to fulfil these functional requirements and is introduced in the following sections.

4.1. Fundamental method

The fundamental idea of the concept is that if a DPC occurs it can be modelled by either adapting the effort invested into design activities or the confidence that designers have into deliverables or both (see Figure 2, modelling DPCs corresponds to requirements 3-5 in Table 1). In addition to these direct effects on singular process elements, there are indirect effects on the iterative behaviour, as it is driven by confidence (or uncertainty, see Table 1, requirement 2), which are represented as follows: The likelihood that an activity triggers iteration depends on the confidence into the deliverables that it consumes. Moreover, the confidence of a downstream deliverable is determined by both the effort invested into the upstream process and the confidence into the upstream deliverables used to create the downstream deliverable. Hence, these two relationships establish a propagation network, in which the confidence into upstream deliverables and the upstream effort determine the confidence into downstream deliverables and thus, the likelihood of iterations of downstream activities, which in turn lead to additional upstream effort and so on. DPCs are propagated through this network and affect both process duration and effort.

This concept can be implemented in ASM, which also supports process analysis through Monte-Carlo simulations (see Table 1, requirements 1 and 8). Moreover, ASM allows the definition of process variables so that change effects can be captured for sub-processes and the holistic process (see Table 1, requirement 6). Finally, the propagation network described above can be examined to suggest alternative DPC reactions and process improvement options (see Table 1, requirement 7).

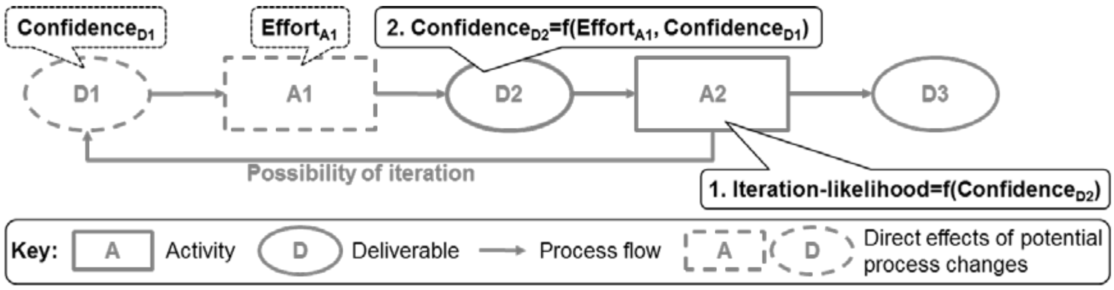

Figure 2. Overall concept for the suggested DPCM.

4.2. Fundamental method elements

The concept is based on two fundamental elements: activities and deliverables (see Figure 2).

Activities are constituent parts of a process, i.e. sub-processes, which can be defined as “packages of work to be done to produce results” (Browning et al. 2006, p.117). They consume inputs and resources, like time, money, people, tools and facilities. Sim and Duffy (2003) suggest a formalism for a generic design activity, which views the activity as converting a design goal and the design agent’s imperfect knowledge into additional knowledge, which may or may not represent a solution to the goal. If the activity does not produce such a solution, a new potentially more manageable goal emerges that prompts alternative actions, which eventually will bring the design-agent closer to a design solution. Sim and Duffy also classify design activities into three major types: definition activities, which reformulate the design problem so that it becomes easier to find design solutions, evaluation activities, which assess potential design solutions, and management activities, which co-ordinate other activities towards a progress in the design solution.

The element of the proposed concept, which is referred to as activity (see Figure 2), is closely aligned with Sim and Duffy's definition and types. The concept particularly relies on two properties of this element: the effort associated with activities and their potential to trigger iterations. In this context, effort comprises the overall cost including knowledge, resources, tools and time that are invested into design activities to transform input into output deliverables and to evolve the design. Furthermore, based on the examined formalism for a generic activity, activities of every major type may trigger iterations.

Given the concept is based on ASM, it assumes that the design can be represented as the generation and refinement of deliverables through activities (see Figure 2). Such a deliverable can be quantitative or qualitative and can describe any characteristic of the product, for example, a parameter like the geometry of a fan blade, a data file like the fan blade's mesh used for stress analysis, or a report like the stress analysis' report (Wynn, Eckert, and Clarkson 2006). Further examples for such deliverables are CAD drawings, bills of material, simulation data or calculation results (Ouertani 2008).

A key characteristic of deliverables utilised by the proposed concept is confidence. Clarkson and Hamilton (2000) based their Signposting modelling framework on the evolution of parameter confidence, which they defined as follows: "To be confident in a parameter means that the parameter is detailed, accurate, robust, well understood, physically realistic and, in the case of a performance parameter, meets pre-defined performance requirements.” This definition of confidence also fits the proposed concept, although one important difference exists: While Signposting grounds on the ‘absolute’ change in confidence caused by the execution of a certain activity (e.g., the confidence into the geometry of an aerofoil increases after a stress analysis compared to the situation before the stress analysis), the proposed method grounds on a ‘relative’ deliverable confidence, which is compared to a usual confidence in this deliverable in similar designs and at a similar design stage (e.g., the confidence into the geometry of an aerofoil is higher than usual after a stress analysis, which was executed with a finer mesh than usual). A relative comparison of deliverable confidence to similar past designs is possible since the proposed method targets mature, evolutionary design processes (see Table 1, requirement 1).

The reason for the use of relative confidence is the proposed dependency of iterative behaviour on deliverable confidence, elaborated in the following section. A relative understanding of confidence allows the same sub-process to produce different levels of confidence in an output, which impacts the iterative behaviour downstream.

4.3. Fundamental relationships between method elements

The first fundamental relationship underlying the proposed method concept is a dependency between the confidence into an activity’s inputs and its likelihood to trigger iterations (see Figure 2), i.e.

$$\text{Iteration} - \text{likelihood} = f(\text{Confidence}_{\text{Inputs}}) \tag{1}$$

It is discussed in the engineering design literature from various perspectives (Lévárdy and Browning 2009; Suss and Thomson 2012; Wynn, Grebici, and Clarkson 2011). Intuitively this relationship is negative, i.e. iteration-likelihoods tend to decrease for high confidence levels in inputs. For example, if a designer has a high confidence into the geometry and the material of a blade it is less likely that a stress analysis activity, which consumes these inputs, will fail and result in the blade's redesign. Inherently the designer's a-priori assessment may be inaccurate and thus, a certain likelihood exists that once executed the stress analysis will trigger iteration.

The influence of input confidence on the iteration-likelihood depends on the specific activity. In fact, there are also activities where iteration-likelihood and input confidence are independent. For example, exploring the design space may be an iterative activity, which is repeated until a certain scheduled duration expires independently from the designer's confidence into identified solutions or requirements. Consequently, the method should be flexible enough to capture different specifications of this relationship. It is noteworthy that this relationship can be only implemented for explicitly captured possibilities of iteration. The modeller thus, needs to carefully choose the model's level of granularity so that relevant iterations are not obscured.

The second fundamental relationship underlying the proposed method concept is a dependency between the confidence into an activity's inputs, the effort invested into the activity and the confidence into the activity's resulting output (see Figure 2), i.e.

$$\text{Confidence}_{\text{Output}} = f(\text{Effort}, \text{ConfidenceInputs}) \qquad (2)$$

This relationship has also been discussed by multiple authors in the engineering design literature (Lévárdy et al. 2003; MacCallum and Duffy 1987; Wynn, Grebici, and Clarkson 2011). Intuitively it is positive, i.e. the confidence into a design activity's output tends to increase with the effort invested into the activity and with an increasing confidence into its inputs. For example, a design team that has a high confidence into the design requirements on hand and spends a substantial time with the generation of alternative concepts, should have a higher confidence into the resulting set of concepts than the same design team if it was not sure about the feasibility of certain requirements and had less time for the concept generation.

Once again, the influence of effort and input confidence on output confidence depends on the specific activity and output. In fact there may be activities that have multiple outputs, of which each is differently affected by the activity's effort and inputs. For example there may be cases where only a sub-set of inputs influences a certain output, or where the output confidence is independent from the effort. The latter case can be well envisioned for certain computational activities, which require a standardised effort so that changes in this effort are infeasible. Thus, the method should be flexible enough to capture such different specifications of this second relationship.

It is noteworthy that an additional network of interactions is introduced through the second relationship, as now activities do not only depend on each other in terms of information precedence constraints but also in terms of confidence levels of their inputs and outputs. Moreover, in their basic form the two relationships have an important interplay: If, for example, a DPC led to insufficient effort devoted to the process upstream, this could result in a lower output confidence, which would increase the likelihood that iterations are triggered downstream. In turn, such iterations would result in an increase in upstream effort so that output confidence would increase and the likelihood of further iterations would decrease. Therefore, in the proposed concept iterations function as a control mechanism for the confidence into the design – an intuitive and intended effect.

4.4. Possibilities of method application

According to the goals of existing DPC support methods (see Section 2), the DPCM is intended to help design teams gain understanding and improve process execution and planning based on the examination of DPC effects. Accordingly, the method should offer a tool box with analysis possibilities, which address each of these points. The following analysis tool box, which is based on the method elements and relationships established in the previous sections, is envisioned as part of the method (see Figure 3):

The analysis to gain understanding comprises the preventive identification of potentially critical DPCs by examining the impacts of confidence decreases in external process inputs or reductions in activity effort (Figure 3, analysis 1) on process duration and effort. The analysis thus, supports risk identification, which is a particularly challenging area within project risk management (Kloss-Grote and Moss 2008). Moreover, this analysis can be also extended to actual DPCs, which are likely to comprise a mixture of input confidence and activity effort changes.

The analysis to improve process execution encompasses the examination of mitigating reactions to reductions in input confidence and/or activity effort (Figure 3, analysis 2) so that an increase of process duration and effort is minimised or avoided.

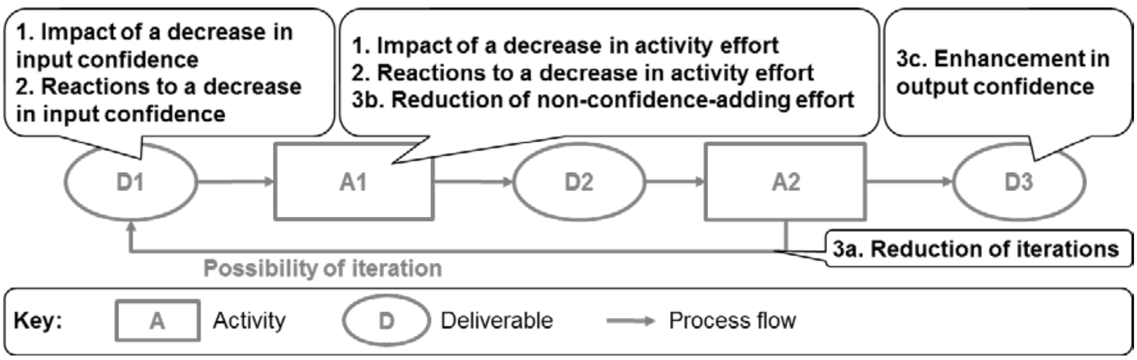

Figure 3. Types of analysis supported by the DPCM.

Lastly, three types of analysis to improve process planning, based on identifying and prioritising the 'right' DPCs, are suggested. The first type examines combinations of confidence increases in specific external inputs and effort investments into specific activities to reduce unwanted iterations and rework and consequently overall process duration and effort (Figure 3, analysis 3a). The second type identifies activities, which should be executed as lean as possible, as additional effort does not increase their output confidence, to reduce overall process effort and duration (Figure 3, analysis 3b). The third type assesses combinations of confidence increases in specific external inputs and effort investments into specific activities to increase the confidence into process outputs (Figure 3, analysis 3c), and therefore, addresses the quality dimension of design process performance.

Overall, the suggested method is thus, envisioned to be used for the investigation of various ‘what if?’ scenarios during design process planning and execution.

5. Detail design of the DPCM

This section examines the DPCM’s detail design and implementation in ASM, and specifies the method’s elements, their relationships and the method’s application.

5.1. Specification of method elements

The detailed specification of activities and deliverables, which are the DPCM’s two fundamental elements (see Section 4.2), is discussed hereafter.

5.1.1. Activities

To recapitulate, the method focusses on two properties of activities: the consumed effort and the potential to trigger iterations (see Section 4.2).

The DPCM differentiates between three discrete effort levels per activity, i.e. low/medium/high (abbreviated l/m/h hereafter), based on a relative comparison to the usual effort of an equivalent activity in similar, past projects. Consequently, two effort level boundaries, which separate medium from low and high from medium effort, need to be defined by the responsible design team per activity. In practice, this presumes the approximation of activity effort through, for example, costs for human labour and the use of equipment or simply the activity's total-execution time multiplied with the number of human resources, measured in person-days.

The design team only needs to define such effort level boundaries for activities, the output confidence of which depends on the invested effort (according to so-called confidence mappings, which are specifications of the second fundamental relationship, see Section 5.2.2). The boundaries should be then specified in such a way, so that a low/high activity effort can be expected to considerably decrease/increase the confidence into the activity's respective outputs. Moreover, as the DPCM builds upon a simulation model of the examined design process, which contains various uncertainties (e.g., probabilistic activity durations and iteration), process simulations of the baseline process before changes can be used to produce frequency distributions of the chosen (cumulative) effort proxy per activity. These frequency distributions can support design teams in defining effort level boundaries, as, for example, the first and fourth quartile of observations per activity could be attributed to low and high effort levels respectively.

Figure 4 shows a screenshot of the implementation of effort levels in ASM. The variable eA1, which is being edited in the screenshot, is used to assign an effort level to the activity A1 during simulations: If the activity's effort, approximated through the activity's total-execution time multiplied with the number of required designers (hrA1), is smaller/larger than the predefined effort boundary emA1/ehA1 the effort is recorded as l/h; if the effort is between emA1 and ehA1 the effort is recorded as m.

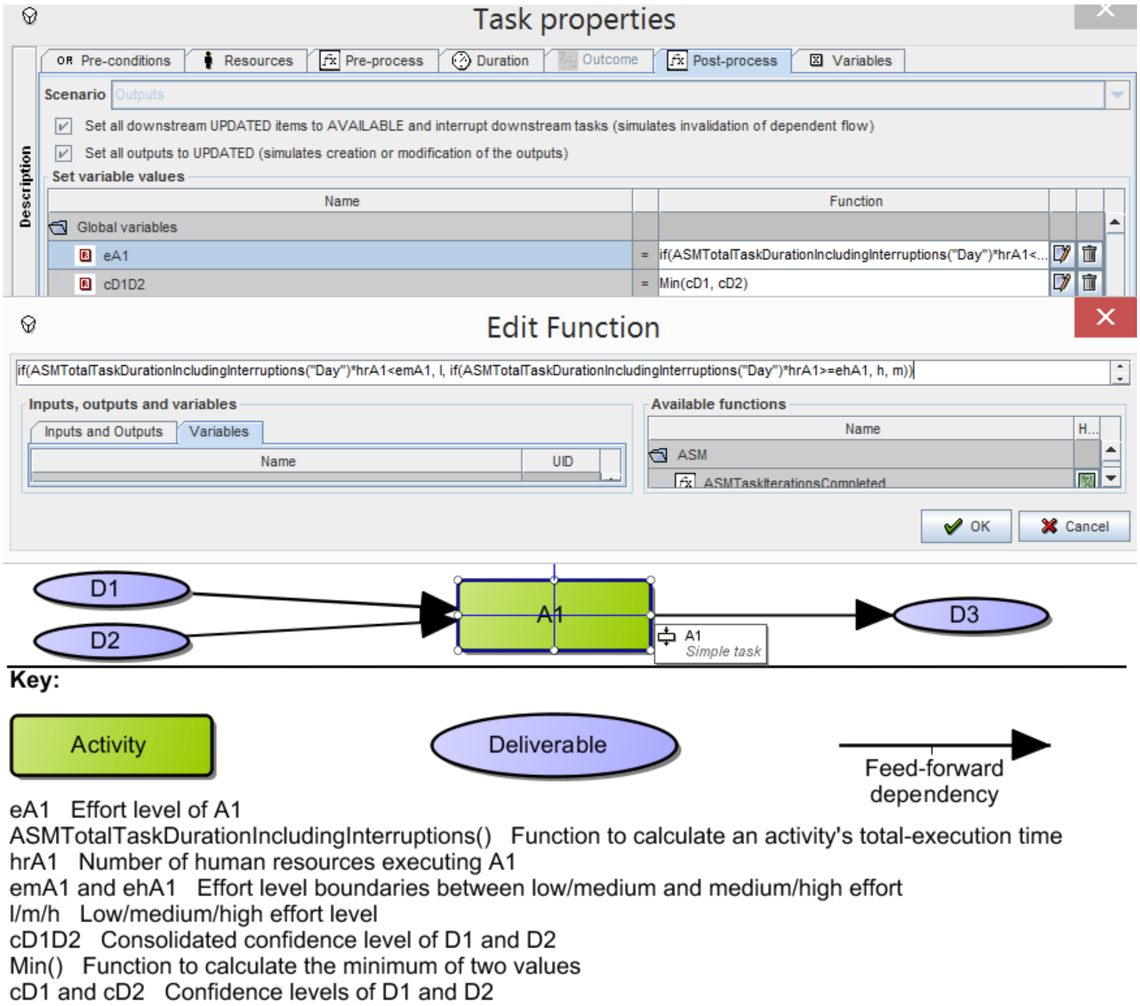

Figure 4. Implementation of effort levels and confidence in ASM.

The DPCM's second relevant property of activities, the causation of iterations, is modelled in ASM through specific constructs, which can be used to represent activities that trigger a single or multiple alternative iteration-loops. The conditions to trigger iteration-loops, can be specified through functions and variables. For example, the outcome of the function rand(), which produces a random variable generated from a uniform distribution between 0 and 1, can be compared to a variable, which denotes a pre-specified iteration-likelihood, to represent probabilistic iterations (see Figure 5).

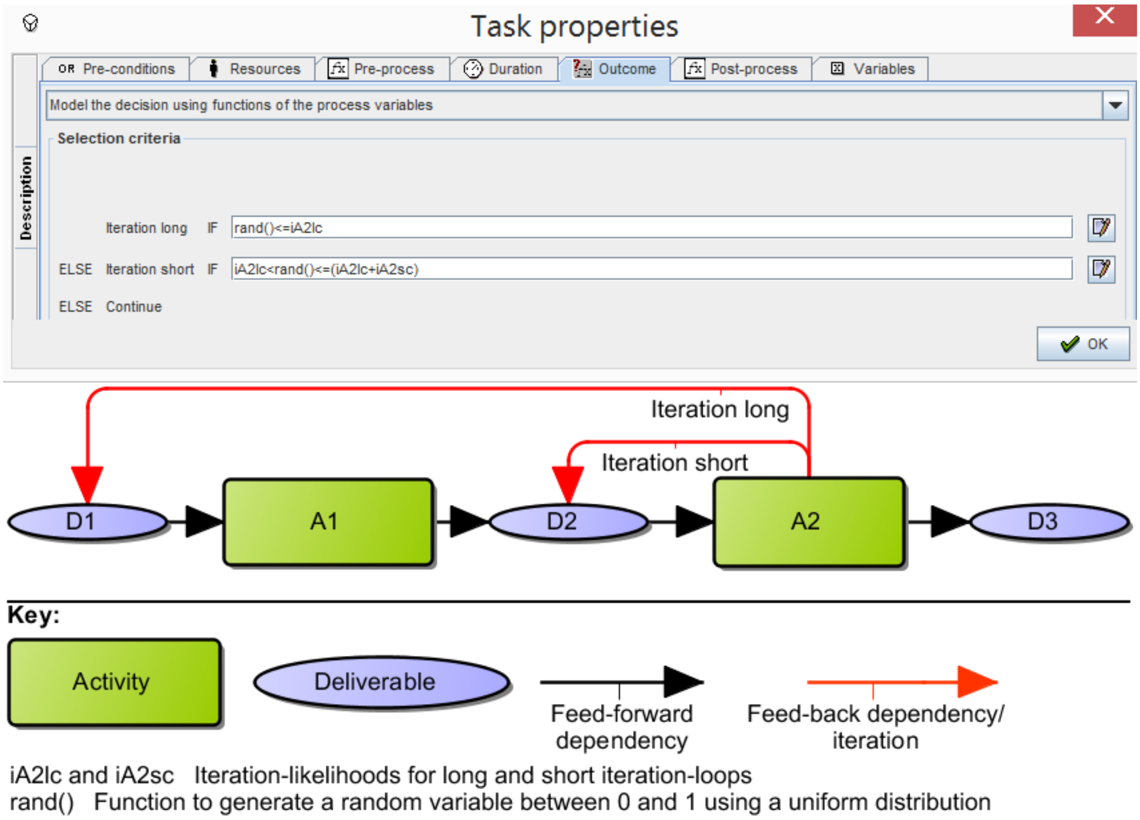

Figure 5. Implementation of iterations in ASM.

5.1.2. Deliverables

Deliverables are the DPCM's second fundamental element (see Section 4.2) and particularly the confidence into deliverables is in the method's focus. Similarly to activity effort, the method differentiates between three discrete confidence levels per deliverable, i.e. low/medium/high (abbreviated l/m/h hereafter), based on a relative comparison to the usual confidence into the deliverable in similar, past design projects and at an equivalent design stage. Applying the DPCM, confidence levels of internal deliverables, which are produced within the examined process, are automatically calculated based on so-called confidence mappings, which are specifications of the second fundamental relationship (see Section 5.2.2). Contrary to internal deliverables, external process inputs come from customers, other upstream or past sub-processes and thus, are not under the direct influence of the examined design process. Per definition if there is no reason to assume a higher or lower confidence into an external input compared to similar designs at an equivalent design stage, a medium confidence level is appropriate. Consequently, confidence levels only need to be specified by the responsible design team for specific external inputs if concrete instances of the design process with changes of the confidence into these inputs are analysed. A low/high confidence level is then assigned to an external input if it can be expected to considerably decrease/increase the confidence into one of its dependent outputs.

As the DPCM's fundamental relationships (see Section 5.2) are used to calculate output-confidence levels and iteration-likelihoods based on input-confidence levels, a consolidated input confidence level needs to be determined for activities with multiple inputs. Based on the case study experience (see Section 6), two alternative consolidation rules, which should be agreed upon with the responsible design team, are suggested for such activities:

(1) Determine the single input with the greatest effect on the confidence into the output, i.e. the 'major input' of the activity, and assume its confidence level as the overall consolidated input confidence level.
(2) Assume the lowest confidence level of all inputs, which influence the confidence into the output, as the overall consolidated input confidence level.

The second consolidation rule is based on the cautious assumption that uncertainties in multiple inputs rather add up and do not cancel each other out. Thus, it is both more conservative and parsimonious with regard to data collection than determining a weighted average of multiple input confidence levels.

Figure 4 also shows the implementation of confidence levels in ASM for an activity with multiple inputs. The independent variables cD1 and cD2 denote the confidence levels of the individual inputs and can be adapted by the modeller to represent changes. The dependent variable cD1D2 determines the consolidated input confidence using a function, which determines the minimum of two values. Thus, cD1D2 consolidates the input confidence according to the second rule outlined above.

5.2. Specification of relationships between method elements

Two fundamental relationships are underlying the DPCM (see Section 4.3): First, a dependency between the confidence into an activity's inputs and its likelihood to trigger iterations and second, a dependency between the confidence into an activity's inputs, the effort invested into the activity and the confidence into the activity's resulting output. The quantitative formalisation of these relationships and their implementation in ASM is discussed in the following.

5.2.1. Input confidence and iterations

To implement the first fundamental relationship in the DPCM, first, an iteration-likelihood i_j^* is estimated for every activity j, which has the potential to trigger iterations, based on the experience of the responsible design team. Then, the design team is asked to specify a multiplicative factor $a_{j,c}$, with $a_{j,c} \geq 0$, for the iteration-likelihood for each possible confidence level c of the activity's consolidated inputs. The iteration-likelihood of activity j is then modelled as follows:

$$i_{j,c} = \mathrm{Min}(a_{j,c} \times i_j, 1) \quad \text{for } c \in \{l, m, h\} \tag{3}$$

Here the Min()-function is introduced to avoid $i_{j,c}>1$. Thus, the multiplicative factors denote a change in the iteration-likelihood depending on the level of input confidence. If $a_{j,l} > a_{j,m} > a_{j,h}$ there is a negative relationship between iteration-likelihood and input confidence, which is the typical case (see Section 4.3). If $a_{j,l} < a_{j,m} < a_{j,h}$ this relationship is positive – a rather theoretic case that was not observed in the case study (see Section 6). Lastly, if $a_{j,l} = a_{j,m} = a_{j,h}$ then the iteration-likelihood of the examined activity and its input confidence are independent. Equation (3) is based on i_j, which is a calibrated iteration-likelihood for the activity, rather than on the specified iteration-likelihood i_j^*. If the equation was based on i_j^* and the process simulations resulted in certain ratios of simulation runs with l/m/h input confidence for the iterative activity depending on the upstream process, it would be purely by chance if these ratios were such, that i_j^* would be replicated overall. Thus, after the definition of $a_{j,c}$ for $c \in \{l, m, h\}$ an i_j needs to be found so that the simulation model is calibrated to produce the specified iteration-likelihood i_j^* on average. Alternatively, i_j could also be specified directly together with the multiplicative factors by the responsible design team. Then, the resulting i_j^* can be derived from simulations and its plausibility should be verified.

The discussed relationship can be implemented in ASM by specifying a condition in the iterative activity`s properties that changes a variable, which describes its iteration-likelihood, depending on the level of input confidence (see Section 5.1). To calibrate the model, it is suggested to run multiple simulation sets with different values for i_j and to converge iteratively to its right value so that the pre-specified i_j^* is produced on average. This requires measuring the resulting average iteration-likelihood for each simulation set and comparing it to i_j^* of the targeted activity j. Usually, if the resulting average iteration-likelihood is too low/high, a higher/lower value of i_j should be tested in the next simulation set. Implementing this procedure in ASM requires the definition of two counting variables per iterative activity to count the number of activity executions with and without iterations respectively. The average iteration-likelihood can then be calculated as the number of activity executions with iteration divided through the sum of executions with and without iterations. If multiple iterative activities are present in the process, this calibration procedure should start with the most downstream iterative activity and then proceed one-by-one to activities further upstream, as downstream iterations affect the effort and number of activity repetitions in the upstream process (Shapiro et al. 2015). Moreover, for activities that can trigger multiple alternative iteration-loops the iteration-likelihoods should be calibrated for all possible loops.

5.2.2. Input confidence, activity effort and output confidence

To implement the DPCM's second fundamental relationship a mapping of input confidence and activity effort to output confidence, called confidence mapping hereafter, needs to be elicited for every activity output in the examined design process. Figure 6 shows three typical mapping possibilities for an activity each represented by a different table. The inside of each table denotes the resulting confidence level of the activity's output for every possible combination of input confidence and activity effort, as shown at the outside of each table. In table a), for example, a medium level of input confidence and a high level of activity effort result in a medium level of output confidence. Thus, while in table a) output confidence depends purely on input confidence (valid, e.g., for certain standardised computational activities), in table b) output confidence depends purely on activity effort (a rather theoretical case) and in table c) output confidence depends on both input confidence and activity effort. The elicitation of such a mapping relies on the experience of the responsible design team. As these mappings need to be consistent with the definition of the respective effort and confidence levels (see Section 5.1), it is important to specify the effort and confidence levels with this application in mind.

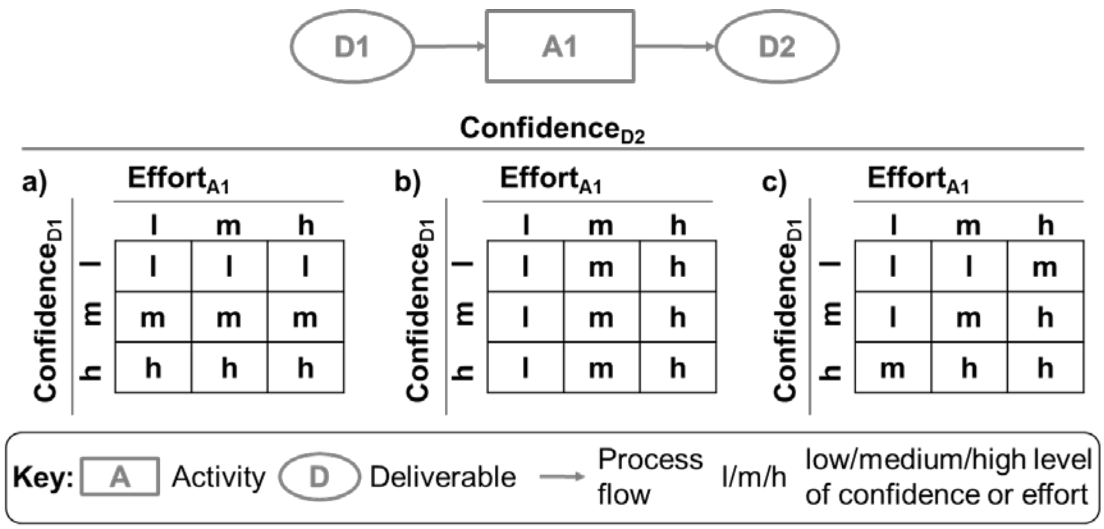

Figure 6. Characteristic confidence mappings for an activity with a single input and output.

Figure 7 shows a screenshot of the implementation of such a mapping in ASM. To determine D2's confidence a dummy activity, which has a symbolic duration of 1 second and is called ConfD2, is created. The mapping is defined in the dummy activity's properties through nested if()-statements, where the variable cD2 denotes the resulting output confidence that depends on the input confidence cD1 and the activity effort eA1, so that the example corresponds to mapping c) in Figure 6. The confidence mapping needs to be implemented in a separate dummy activity because ASM does not understand the correct execution order of first defining and then operating on variables in the same properties tab. This circumstance thus, requires a doubling of the activities on the process map. To maintain readability of the model it is suggested to group each activity and the respective dummy into a sub-process (yellow frame in Figure 7). Such sub-processes can be collapsed so that dummy activities are not visible if not required.

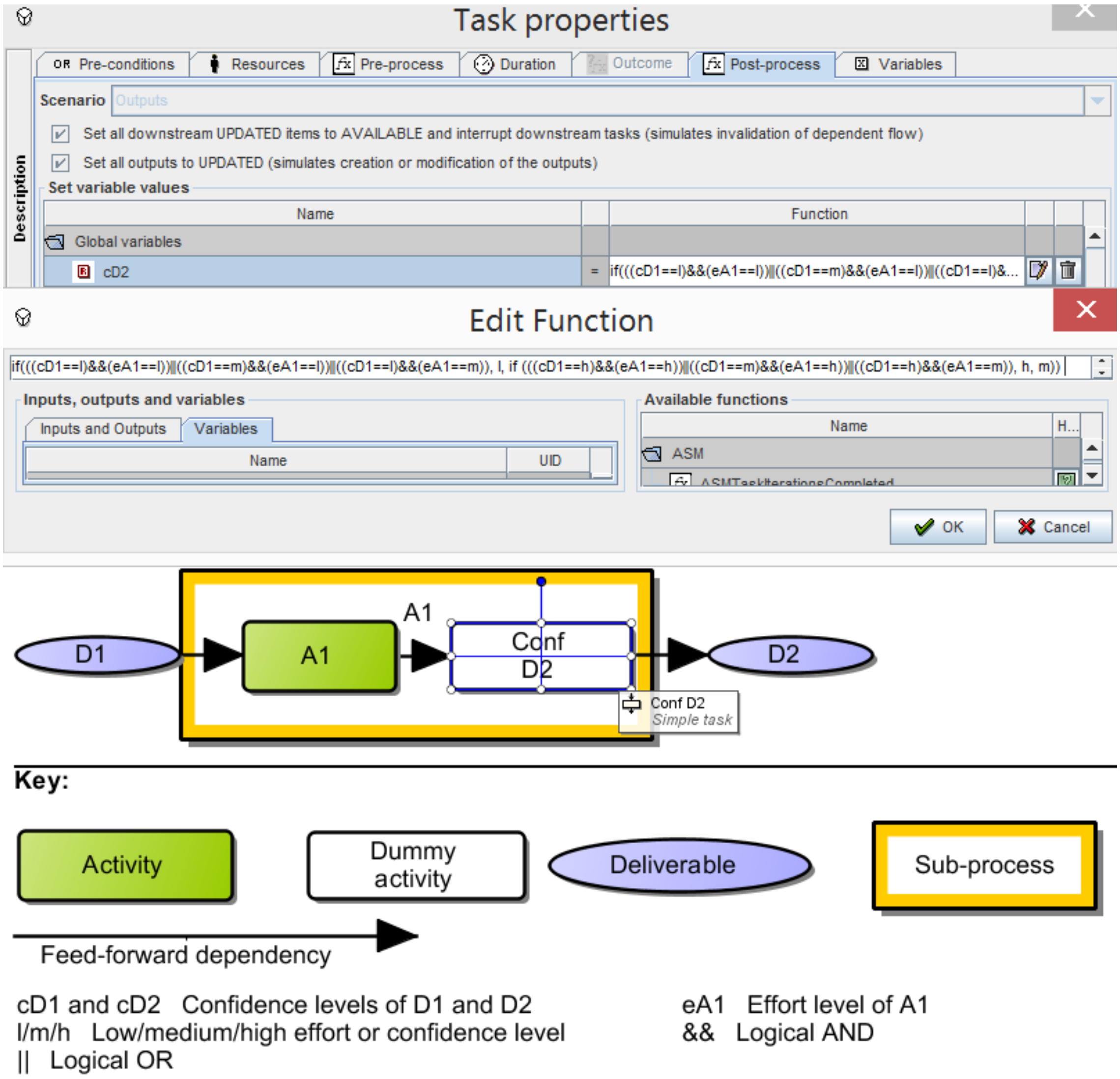

Figure 7. Implementation of confidence mappings in ASM.

5.3. *Discussion of model building*

This section discusses the DPCM model building as described throughout the specification of method elements and relationships in Sections 5.1 and 5.2. First, the effort for collecting the required data and incorporating it in a DPCM model is examined, before possibilities to verify the model’s accuracy are discussed.

Provided that a process flowchart, which shows the activities, deliverables and the process flow of the examined design process, exists or has been developed, there are two types of input data required to apply the DPCM:

(1) Basic process modelling data, comprising a duration (probability distribution), resource requirements (human, computational and other resources) and an iteration-likelihood (if applicable) for each activity; If other effort proxies than the total-execution time of activities multiplied with the number of utilised resources are available and relevant, these should be also specified through probability distributions per activity;
(2) DPCM-specific data, comprising effort level boundaries, input confidence consolidation rules, confidence mappings, multiplicative factors for iteration-likelihoods, and adapted effort and confidence levels for specific change cases.

In the reported fan sub-system case study the basic data was collected during a six-hours workshop (see Section 3), of which approximately four hours were spent on the examined core part of the process with 27 activities (see Figure 9). Additionally, the relevant DPCM-specific data was collected during another half-day workshop. Moreover, building the basic ASM model after the first workshop took approximately five days and enriching it with the DPCM-specific data after the second workshop required another two days. Although this model building effort depends on the specific process, it can be expected to be approximately proportional to the number of activities in the underlying process flowchart. Also, it is noteworthy that the developed model can be reused for applications of the DPCM to future executions of the examined design process, unless the complete process is restructured or the methods and tools utilised are altered substantially. Furthermore, the first type of data is useful for design process management irrespective of the DPCM, as it is provides fundamental information for process planning and scheduling.

The accuracy of the developed model should be ensured in a number of ways. First, all data should be collected with at least two design team members providing an opinion and jointly agreeing on a value, which reduces the subjectivity of results compared to interviewing a single person. Second, the basic process modelling data should be verified through running process simulations of the basic ASM model to produce frequency distributions of the overall process duration and effort. The plausibility of these process performance measures should then be assessed based on the team's experience with past executions of the examined or similar design processes. Lastly, as the ASM model that is enriched with DPCM-specific data is calibrated to replicate the baseline process performance (see Section 5.2.1), the accuracy of the DPCM-specific data only needs to be verified for the representation of changes. This could be done by examining and comparing the effects of DPCs in past design processes to predictions of the DPCM, which presumes that past design processes were documented accordingly, including a differentiation between direct and indirect change effects and a record of initiated reactions to DPCs. As this is rarely the case and also because of the DPCM's intended applicability to hypothetical changes, like the identification of potentially critical DPCs or process improvements (see Section 4.4), it is rather suggested that the plausibility of respective results is discussed with the design team. The team should then be able to assess whether the activities and deliverables identified as potentially critical or relevant for improvements are reasonable based on its expertise and experience.

While it is acknowledged that the described procedure does not ensure an entirely objective evaluation of the model's accuracy and the related accuracy of analysis results – a fundamental difficulty that is also well-known in the evaluation of EC-propagation methods (Hamraz 2013) – the listed steps are deemed adequate to ensure useful results, which can be confidently utilised by the design team according to the authors' case study experience.

5.4. *Specification of method application*

Due to space constraints it is not possible to discuss the detailed implementation of all analysis types enabled through the DPCM (see Section 4.4). However, as all of these analyses can be realised through a similar sensitivity study procedure, first this procedure is explained hereafter. Then, because of its complexity and noteworthy application results, analysis 3a (see Figure 3), which supports process planning by reducing unnecessary iterations, is detailed.

5.4.1. Sensitivity study procedure underlying the DPCM's analyses

The sensitivity study procedure underlying most of the DPCM's analyses is similar to one suggested earlier by the authors (Shapiro et al. 2015). It is based on comparing the design process effort and duration as well as the effort and duration of singular activities before and after a change or improvement. Two particularly important measures for this procedure are an activity's total-execution time t_n, which is defined as its cumulative duration over all repetitions during the process (due to rework), and an activity's total effort te_n, which is quantified as costs or can be approximated through the activity's total-execution time multiplied with the number of designers required for its execution, as described in Section 5.1.1.

The sensitivity study procedure can then be summarised as follows:

(1) Run a first set of Monte-Carlo simulations on a DPCM model of the design process of interest. This set simulates the baseline case with initial deliverable confidence and activity effort levels before any change or improvement. The model requires the definition of variables to measure 1) the total-execution time t_n and 2) the effort te_n for each activity n=1, …, N as well as 3) the process effort TE and 4) the process duration D.

(2) Obtain estimates for 1) the expected total-execution time $E(t_n)$ and 2) the expected effort $E(te_n)$ for each activity n=1, …, N as well as for 3) the expected process effort $E(TE)$ and 4) the expected process duration $E(D)$. This requires calculating 1) the average total-execution time and 2) the average effort for each activity as well as 3) the average process effort and 4) the average process duration over all simulations respectively.

(3) Model a process change or improvement (i.e. change confidence levels of deliverables and/or effort levels of activities) and run a set of Monte-Carlo simulations of the adapted design process. Repeat this for every of the M change or improvement constellations of interest, so that M simulation sets are run overall.

(4) For each of the M simulation sets repeat step 2 thus, resulting in estimates for 1) $E(t_{n,m})$, 2) $E(te_{n,m})$, 3) $E(TE_m)$ and 4) $E(D_m)$ with m=1, …, M.

(5) For each of the M simulation sets calculate estimates for relative changes in 1) the expected total-execution time and 2) the expected effort per activity as well as in 3) the expected process effort and 4) the expected process duration, i.e. 1) $\Delta e(t_{n,m})$, 2) $\Delta e(te_{n,m})$, 3) $\Delta e(TE_m)$ and 4) $\Delta e(D_m)$ for m=1, …, M.

The highest absolute values of $\Delta e(TE_m)$ and $\Delta e(D_m)$ indicate the most critical changes or most effective improvements and the design manager might need to trade-off these two process performance dimensions, i.e. effort and duration, when assessing the criticality or effectiveness of changes. Moreover, the measurement of $\Delta e(t_{n,m})$ and $\Delta e(te_{n,m})$ enables identifying the activities, which are affected the most by each change or improvement.

The required simulation sets for different change and improvement constellations can be carried out in ASM, which includes a functionality for simulation experiments that allows to define a different value for each independent variable per simulation set. As confidence levels of external process inputs are described by independent variables (see Section 5.1.2), they are straight-forward to control. However, because activities accumulate effort during the process, activity effort levels cannot be directly described by independent variables. Instead, the DPCM compares the accumulated effort per activity to its defined boundaries and calculates effort levels accordingly (see Section 5.1.1). Thus, an independent control variable and additional conditions need to be defined so that the required effort level is reached per activity execution, as demonstrated in Figure 8: Given a certain number of human resources (hrA1) involved in the activity A1, the variable econA1 controls the activity's duration to produce a required effort level. It is important that the control variable's value is reset after activity completion if an activity is not meant to be repeated with a certain effort level in case of rework.

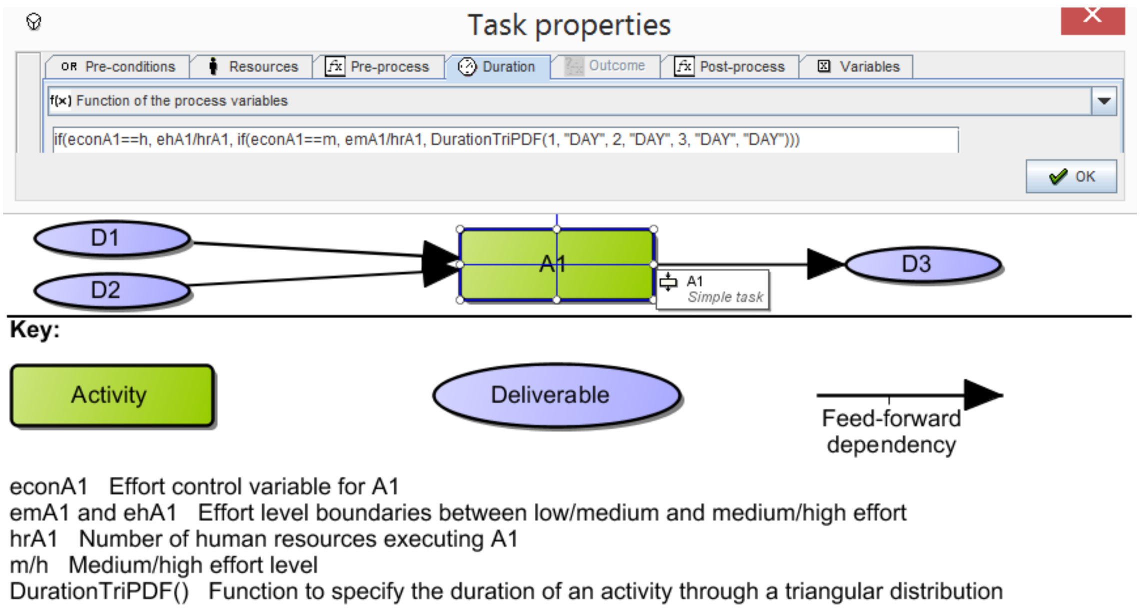

Figure 8. Use of variables to control the effort levels of activities during simulations in ASM.

5.4.2. Support of process planning through reduction of iterations

This analysis (3a in Figure 3) examines the impacts of combinations of input confidence and activity effort increases on the iterative behaviour of the design process. Fundamentally, a design process could be improved through the investment of effort into the right activities within the process or within upstream sub-processes that create relevant process inputs. By doing so confidence levels of internal deliverables are increased and unnecessary iterations are reduced, so that the overall process effort and duration may be decreased, although more effort is invested into certain areas.

Although one-at-a-time increases in deliverable confidence or activity effort levels could be easily examined, such improvements may not be effective as depending on their specific confidence mappings (see Section 5.2.2) activities may require certain confidence and effort level combinations to produce high confidence outputs to reduce downstream iteration. Thus, before applying the sensitivity study procedure described above, sensible improvement combinations need to be identified: If the baseline case before changes is analysed for a design process it is, per definition, sensible to assume medium confidence levels of external process inputs (see Section 5.1.2). Improvement combinations will thus, contain medium and high confidence levels of external inputs. Contrary, even if the baseline case is considered effort levels could be low for the first-time execution of activities, because effort accumulates during the process as activities are reworked until it reaches medium levels on average. Hence, improvement combinations will contain activity effort increases to medium or high levels, but only for the first-time execution of activities. As effort accumulates, the positive effect of an effort increase in an activity's first-time execution on its output confidence is sustained if rework occurs. Moreover, there are usually external inputs and activities, which are irrelevant and can be neglected for this analysis, as they do not affect the confidence levels of internal deliverables, which may affect relevant downstream iterations. Such deliverables and activities can be easily excluded based on the process structure and the collected confidence consolidation rules and confidence mappings.

Overall, the improvement combinations will thus, increase confidence levels of relevant external inputs to high and effort levels of relevant activities to medium or high. However, as not all inputs and activities will be improved at the same time, the combinations will contain l_c=2 confidence and l_e=3 effort levels. Even for a relatively small design process model with i=10 relevant external inputs and a=20 relevant design activities this would result in $l_c^i \times l_e^a = 2^{10} \times 3^{20} \approx 3.6 \times 10^{12}$ possible combinations if a full-factorial design of experiments (DoE) was applied. As evaluating such a large number of improvement combinations through simulations is computationally not feasible, a sub-set of these combinations needs to be selected. Such a selection can either be automated by applying more parsimonious DoE schemes or conducted manually by applying a decision-tree analysis of the most promising improvement combinations.

Various DoE schemes are available, which differ in the number of combinations that are tested and which are applicable in different situations depending on the characteristics of the factors (here: external inputs and activities) and the response surface (here: process duration and effort). Based on a comparison of common experimental designs conducted by Sanchez and Wan (2012), random Latin hypercube designs or 512-design point nearly orthogonal and balanced mixed designs (512 NOB; Vieira et al. 2011), which are both general-purpose designs applicable to models with a large number of interacting factors with different numbers of levels, are suitable to identify improvement combinations of increases in input confidence and activity effort. Contrary to 512 NOB, which prescribes 512 design points, i.e. improvement combinations, random Latin hypercube designs allow the specification of the number of design points to be examined. More design points increase the chance of identifying improvement combinations with greater reductions in process effort and duration, but also increase the computational effort for simulations. DoEs can be automated through widely available tools and thus, are convenient to identify improvement combinations.

Alternatively, a selection of promising improvement combinations can be conducted manually through decision-tree analysis, but requires a good understanding of the DPCM. This analysis begins downstream at the activity that triggers a critical iteration. Usually, its iteration-likelihood decreases for a high confidence level of its input (see Section 4.3). Consequently, the user examines the upstream activity, which produces this input (its own output), and analyses the conditions to generate a high confidence level output based on the confidence mapping (see Section 5.2.2). The user could, for example, find that the upstream activity requires a 1) medium input confidence and high effort level or 2) high input confidence and medium effort level or 3) high levels of both input confidence and effort to produce an output of high confidence. Then, the user would continue to examine the conditions of the activity further upstream to produce the first upstream activity's input according to the required confidence levels and so on. Thus, going backward through the process the user will identify multiple improvement combinations of increases in deliverable confidence and activity effort, which are effective as they result in the input confidence level that minimises the downstream activity's iteration-likelihood. The number of these effective combinations can be further reduced through filtering out non-efficient improvement combinations, which produce the required confidence levels in internal deliverables but use higher effort and/or higher confidence levels than necessary. In the above example option 3) is non-efficient as option 1) requires a lower input confidence level and option

2) requires a lower activity effort level, but all three options lead to the same output confidence. Moreover, it is suggested to avoid improvement combinations which result in low confidence levels of internal deliverables, as these could be inputs to other iterative activities and potentially trigger different iterations.

While the suggested DoE schemes can be applied to identify improvement combinations based on process models with overall up to 300 factors (Sanchez and Wan 2012), i.e. sum of relevant external inputs and activities, the decision-tree analysis is only feasible for relatively small design process models. In fact, producing the decision-tree for reducing iterations in an activity of the process examined in the case study (see Section 6.2) took approximately three hours, examining a model with 18 factors. Although the effort to build such decision-trees depends on the specific design process at hand, it can be expected to grow exponentially with the size of the examined process model. Models with >35 factors should thus, be analysed using DoEs as building corresponding decision-trees could already take longer than a working day.

Once promising improvement combinations are identified a simulation experiment, which corresponds to the procedure described in Section 5.4.1, needs to be defined and executed, where each simulation set corresponds to an improvement combination of certain input confidence and activity efforts levels. Then, the impacts of the improvement combinations on overall process effort and duration are compared and the most effective combinations should be assessed for their practical feasibility before being selected for implementation.

6. Application of the DPCM

The DPCM was applied to the fan sub-system preliminary design process for civil engines at Rolls-Royce PLC. The following sections first elaborate on the case study’s background and subsequently present the application results. Although, each analysis type discussed in Section 4.4 was applied to the fan sub-system design, the reported results focus on the analysis to support process planning through reduction of iterations as detailed in Section 5.4.2. This focus is chosen due to this article’s space constraints and also because of the particularly interesting results of this analysis.

6.1. Case study background

The fan sub-system preliminary design process aims at submitting a bid that contains the weight, the cost and the aerodynamic efficiency of a mechanically acceptable fan sub-system to the whole system design team, which is responsible for the preliminary design of the overall jet engine. This design process is illustrated using its ASM model (see Figure 9), which shows design activities as green rectangles, verification activities as yellow rectangles, deliverables as white ellipses, feed-forward dependencies as black arrows and feed-back dependencies (iterations) as red arrows.

The process begins with the generation of different concepts for the fan blade based on functional requirements, which correspond to input P1D1 in Figure 9, from the whole system design team. Subsequently, one of the resulting concepts is selected to produce the blade geometry, which is then subject to aero-thermal design activities. These are followed by mechanical design activities, which comprise stress and impact analyses of different fidelities as well as a manufacturing assessment. Once the last of these analyses, i.e. activities V22 and V23 in Figure 9, are successfully completed, weights, costs and aerodynamic efficiencies of the fan components are calculated and submitted as a bid to the whole system design team. It is noteworthy, that the calculation of these bid attributes is not shown in Figure 9, as the respective activities were not in the focus of the analyses presented hereafter. This is because all of the major changes and iterations usually occur in the illustrated core part of the process, i.e. before the calculation of bid attributes.

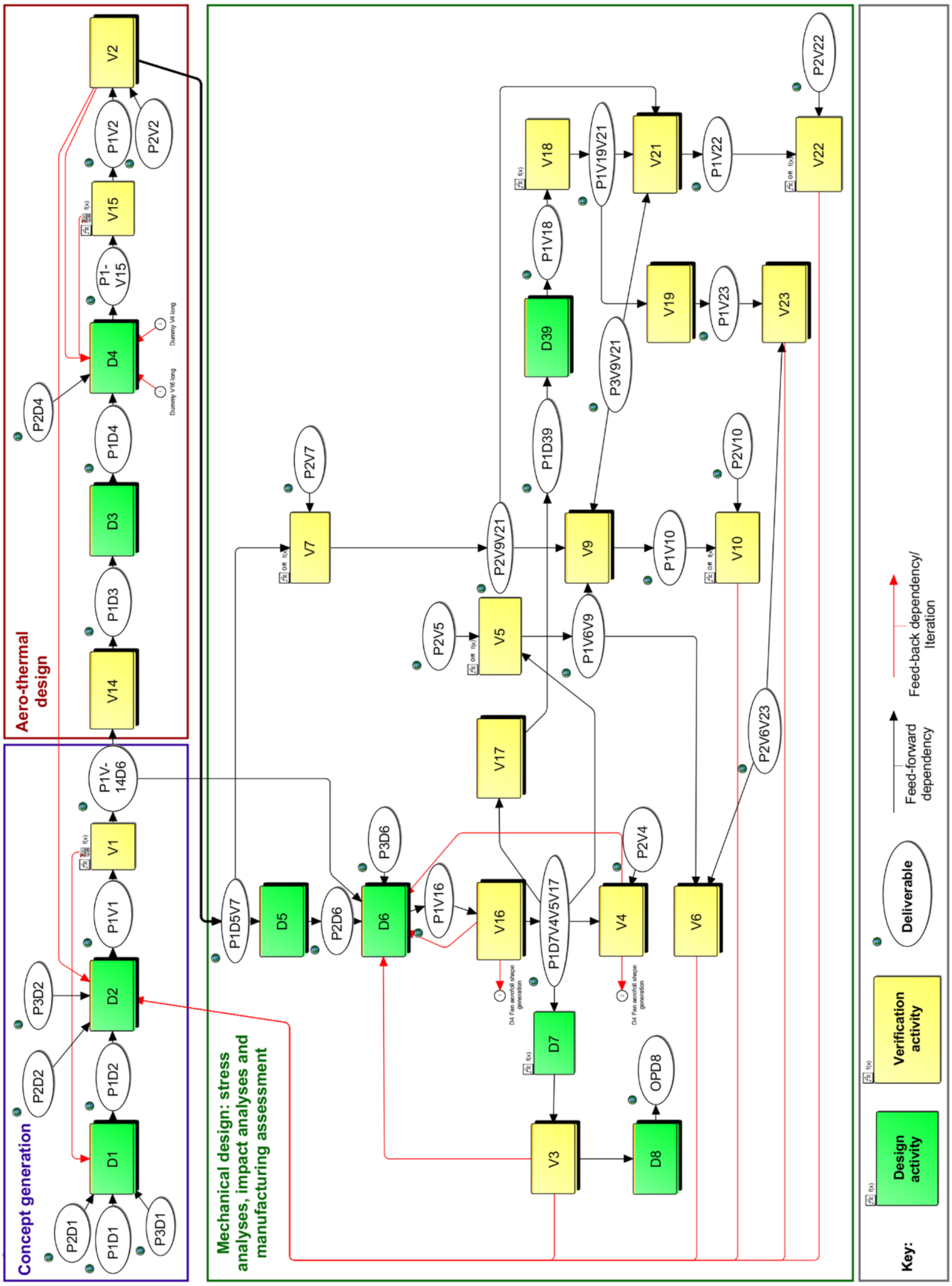

Figure 9. ASM model of the fan sub-system preliminary design process.

Although the descriptions of activities and deliverables are replaced through numbers and letters in Figure 9 to maintain confidentiality, the following characteristics should be emphasised to increase the reader's understanding of the examined process:

(1) The process comprises design and verification activities in the areas of concept generation, aerothermal and mechanical design;
(2) Most of the activities are conducted by a single designer over a duration of a couple of hours to a couple of days, utilising different types of desktop software;
(3) Some activities have significant waiting times as they require shared resources, for example high-performance computers;
(4) Many verification activities have high likelihoods to trigger long iteration-loops;
(5) Overall, the process takes several months, involving more than ten designers from different departments.

While being sufficiently complex due to the described characteristics the process is also well understood as at least parts of it are conducted regularly for different designs. It is thus, particularly suitable for the application of a novel method such as the DPCM.

Initial interviews and ASM process simulations after the first data collection workshop (see Section 3) led to the identification of some key drivers of process performance, including the occupation of designers with parallel projects, waiting times for shared resources, concurrent execution of process parts and iterations triggered through verification activities. The design team's large interest in the latter, i.e. exploring the sources of iterations and possibilities to reduce these, was a strong motivation for the DPCM's development and the analysis presented hereafter.

6.2. Analysis results

Initial simulations of the fan sub-system preliminary design process showed that iteration triggered by the activities V22 and V10 (see Figure 9) are particularly critical for the process, causing together approximately 40% of the overall process duration. Thus, possibilities to reduce the iterations of these activities were examined based on the procedure described in Section 5.4.2.

For this, the baseline scenario before changes and improvements was simulated with 50,000 and each of the identified improvement combinations, discussed hereafter, was simulated with 20,000 process runs. Moreover, activity effort was approximated through the activity's total-execution time multiplied with the number of designers required for its execution and was thus, measured in person-days.

The choice of the number of simulation runs is a trade-off between simulation time, which is proportional to the number of runs, and accuracy, which depends on the uncertainty present in the underlying simulation model. This uncertainty comes from the probability distributions of activity durations and effort as well as the possibility of iterations, and can be quantified by the variance of simulation results. For the examined design process and the chosen number of simulation runs the required simulation time was approximately four hours for the baseline scenario and 1.5 hours for each of the improvement combinations on a 3.6 GHZ desktop computer. The respective analysis results, presented in the following, are considered to be reasonably accurate: The 95%-confidence intervals for the overall process duration and effort range less than $\pm 0.3\%$ around the average values of simulation results for the baseline scenario and less than $\pm 0.5\%$ for each of the improvement combinations, due to the smaller number of process runs utilised for the latter. To reduce these errors even further the number of simulation runs can be increased. However, for such Monte-Carlo simulations the errors are of order $n^{-0.5}$. Thus, if the simulation runs are increased by a factor of 100 the errors will only decrease by a factor of 10. Overall, due to the high uncertainty present in the examined design process, the moderate simulation time and the relatively accurate results, it can be inferred that the developed method is also appropriate for design processes with even greater uncertainties.

To identify combinations of confidence and activity effort increases, which could reduce iterations in each of the two activities V22 and V10, the manual decision-tree analysis (see Section 5.4.2) was conducted. This resulted in 20 different improvement combinations targeting V22, of which 16 combinations produced a high consolidated confidence level of V22`s inputs and four combinations produced a medium confidence level. Moreover, nine improvement combinations targeting V10 were identified, eight producing a high consolidated confidence level in its inputs and one producing a medium confidence level. The combinations resulting in a medium confidence level were included to examine potential improvements, which do not rely on confidence increases in external inputs and thus, are realisable purely from within the fan sub-system preliminary design process.

Figure 10 provides a summary of the improvement combinations for V22, indicating which activities should be executed with additional effort and which external inputs should be increased in confidence as well as showing their average simulated effects. Although all combinations considerably decrease V22's iteration-likelihood, some are predicted to be more effective than others.

For example, combination #20 is predicted to decrease V22's iteration-likelihood, but also to decrease the overall process performance and thus, the additional effort invested may not pay off. Combination #1 is predicted to exhibit the strongest impact, reducing both overall process duration and effort by approximately 20%.

#	Input confidence P3 D1	P2/P3 D2	P2 D4	P3 D6	P2 V7	P3V9 V21	P2 V22	Activity effort D2	D6	D39	V2	V18	V21	Count high confidence	Expected iteration-likelihood V22, relative to baseline	Expected process duration, relative to baseline	Expected process effort, relative to baseline
No change	m	m	m	m	m	m	m	l	l	l	l	l	l/m	0	100%	100%	100%
1	h	h	h	h	h	h	h	m	m	m	m	m	m	7	45%	79%	81%
2	h	h	m	m	m	m	h	l	h	l	m	m	h	3	54%	82%	82%
3	h	h	m	m	m	m	h	l	h	m	m	l	h	3	49%	82%	83%
4	m	m	h	h	h	h	h	h	m	m	m	m	m	5	44%	84%	85%
5	m	m	m	m	m	m	h	m	h	l	m	m	h	1	54%	84%	84%
6	m	m	m	m	m	m	h	m	h	m	m	l	h	1	50%	85%	85%
7	h	h	h	h	h	h	h	m	m	l	m	h	m	7	44%	85%	86%
8	h	h	m	m	m	m	h	l	m	m	m	m	h	3	47%	86%	87%
9	m	m	m	m	m	m	m	m	h	l	m	m	m	0	81%	88%	89%
10	m	m	m	m	m	m	h	m	m	m	m	m	h	1	46%	88%	89%
11	m	m	m	m	m	m	m	m	h	m	m	l	m	0	81%	90%	91%
12	h	h	h	h	h	h	h	m	l	m	m	h	m	7	44%	90%	90%
13	m	m	h	h	h	h	h	h	m	l	m	h	m	5	45%	90%	90%
14	h	h	h	h	h	h	h	m	l	h	m	m	m	7	44%	94%	93%
15	m	m	m	m	m	m	m	m	m	m	m	m	m	0	80%	94%	95%
16	m	m	h	h	h	h	h	h	l	m	m	h	m	5	44%	96%	94%
17	h	h	m	m	m	m	h	l	m	h	m	l	h	3	47%	96%	94%
18	h	h	m	m	m	m	h	l	m	h	m	l	h	3	46%	98%	96%
19	m	m	m	m	m	m	h	m	m	h	m	l	h	1	44%	99%	97%
20	m	m	h	h	h	h	h	h	l	h	m	m	m	5	79%	104%	102%

Key: l: Low level of confidence/effort m: Medium level of confidence/effort h: High level of confidence/effort Efficient combination

Figure 10. Effects of improvement combinations to reduce iterations of V22; improvement combinations ordered by increasing expected process duration.

However, it relies heavily on confidence increases in external process inputs, which are not controllable from within the process. To account for this, combinations that lead to greater improvements than others, but rely on a smaller or equal number of high confidence external inputs are marked in grey and called efficient combinations (the combinations #2/#3 and #5/#6 are marked because the simulated effects of these combinations are comparable). The best performing combination that does not rely on confidence increases in external inputs, is #9, resulting in expected process duration and effort reductions of more than 10% purely through adding effort into the right activities. It is noteworthy that the improvement effects of the combinations are not purely due to less iterations in V22. As V22 is among the last activities in the design process (see Figure 9) multiple other iterative activities further upstream are benefiting from the propagating confidence increase induced into the process by the combinations.

The manually identified improvement combinations to reduce iterations in V22 were also compared to combinations that were automatically generated through the two

DoE schemes introduced in Section 5.4.2, i.e. NOB 512 with 512 design points and random Latin hypercube with 100 design points, the most effective combinations of both of which resulted in estimated process duration reductions of 16%. Although these schemes examined a higher number of combinations and consequently required more computational effort, their suggested combinations thus, resulted in smaller predicted improvements in process performance compared to the combinations identified through decision-tree analysis (16% versus 21% reduction in process duration comparing the best cases). Even though, the results of the DoE schemes are, compared to the decision-tree analysis, somewhat less effective, their advantage is that they neither require a detailed method understanding nor a time-consuming manual investigation of the design process (see Section 5.4.2). In industrial practice the identification of potential improvement combinations for larger design process models should be thus rather carried out using automated DoE schemes.

The improvement combinations to reduce iterations of V10 were analysed similarly and the most effective combination is predicted to improve overall process duration and effort by up to 18%. As for V22 efficient improvement combinations were selected based on the number of required external inputs with an increased confidence. Then, to identify opportunities that enhance process performance even further, each of the efficient improvement combinations for V22 was combined with each of the efficient ones for V10, resulting in 22 joint improvement combinations, where each joint combination is comprised of the highest confidence and effort level of each deliverable and activity that is included in one of the two separate combinations. As expected, the joint improvement combinations could reduce overall process duration and effort even more by up to 29% (see Figure 11). Remarkably, combination #22 is the only one, which does not rely on confidence increases in external inputs and may be implemented purely through adding effort within the fan sub-system preliminary design process.

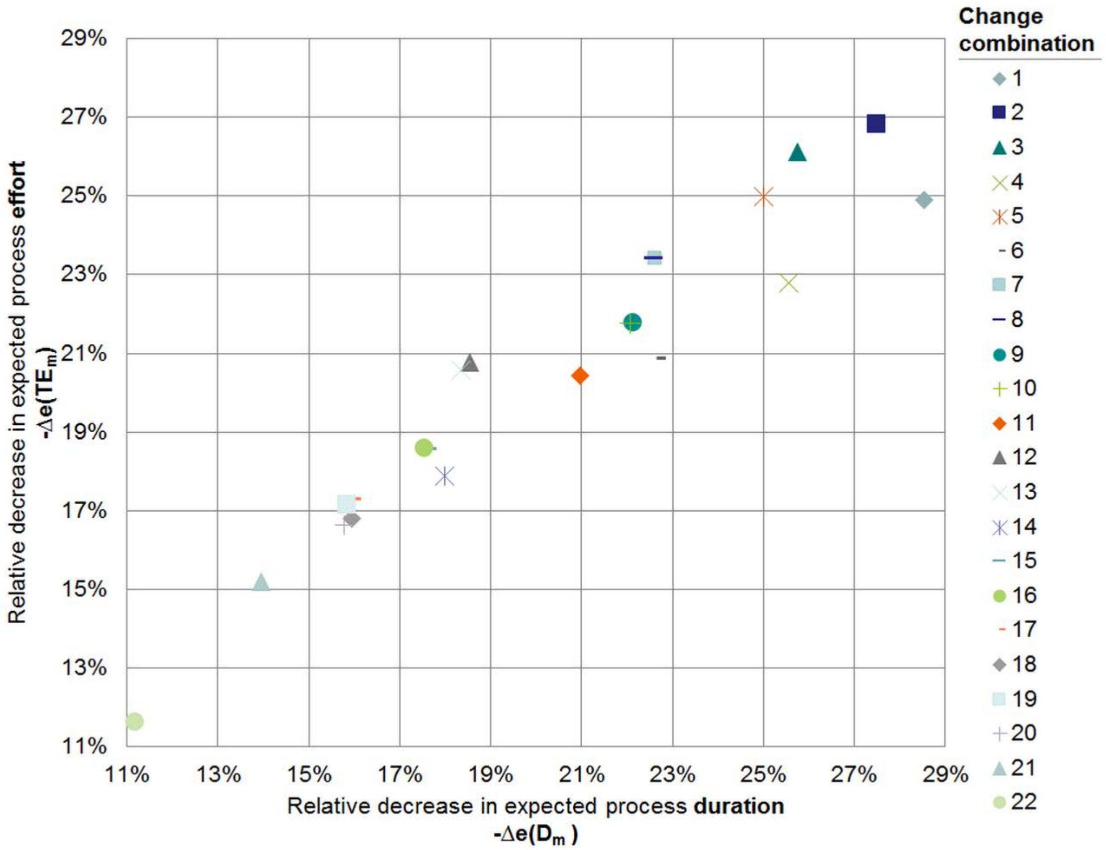

Figure 11. Effects of joint improvement combinations to reduce iteration-likelihoods of V10 and V22; improvement combinations ordered by increasing expected process duration.

To increase the results' practical insights, also the effects on individual activities can be examined. The heat-map in Figure 12 illustrates these effects for the 22 joint improvement combinations as expected percentage changes in each activity's effort, where green/red signifies decreases/increases in an activity's effort. The values for the total process in the last line are the same as in Figure 11 and indicate that all combinations lead to decreases in overall effort. Nevertheless, additional effort investments are required in certain activities in almost every improvement combination, particularly in D6, V9 and V21. Contrary, there are many activities like V16, D5 and D7, which require significantly less effort than originally throughout the improvement combinations. Thus, the figure provides a useful tool for design process management, showing information on the changed effort and resourcing requirements per activity for any given improvement combination.

Affected activity n	**Relative change in expected effort per activity for each improvement combination** Δ**e(te$_{n,m}$)**, in percent																					
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**ø**
D1	-20	-19	-19	-19	-19	-16	-16	-16	-16	-15	-16	-16	-16	-13	-15	-16	-12	-14	-12	-13	-13	-9	**-16**
D2	-14	-14	-13	-13	-14	-7	-5	-5	-4	-4	-6	-5	-5	3	-4	-4	3	2	4	2	3	11	**-5**
D3	-46	-46	-45	-46	-46	-40	-37	-37	-37	-37	-38	-37	-37	-30	-37	-37	-29	-31	-29	-31	-30	-22	**-37**
D4	-43	-43	-42	-42	-42	-39	-34	-33	-33	-34	-39	-34	-33	-30	-34	-33	-30	-30	-30	-31	-30	-26	**-35**
D5	-53	-53	-52	-52	-53	-47	-51	-51	-50	-50	-46	-51	-51	-44	-50	-50	-44	-45	-44	-45	-45	-38	**-49**
D6	61	-10	62	-10	61	-5	63	63	63	63	67	62	63	68	63	63	69	68	69	68	68	74	**54**
D7	-44	-48	-46	-47	-47	-41	-41	-42	-41	-42	-39	-45	-45	-35	-45	-45	-37	-39	-38	-39	-38	-30	**-42**
D8	-25	-28	-26	-25	-27	-23	-21	-21	-22	-21	-14	-21	-21	-9	-21	-21	-16	-19	-16	-19	-8	-4	**-20**
D39	-7	-13	-47	-12	-12	-5	-41	-41	-4	-4	-39	-46	-46	-33	-9	-9	-38	-3	-37	-3	-38	-30	**-23**
V1	-48	-49	-47	-48	-48	-42	-39	-39	-39	-39	-40	-39	-39	-32	-39	-39	-31	-33	-31	-33	-32	-24	**-39**
V2	-5	-5	-5	-5	-5	-1	6	6	6	6	0	6	7	11	7	7	11	11	12	10	11	16	**4**
V3	-44	-45	-44	-44	-44	-38	-41	-41	-41	-41	-36	-43	-42	-34	-42	-42	-35	-36	-35	-36	-35	-27	**-40**
V4	-43	-44	-43	-43	-43	-38	-41	-41	-41	-41	-36	-42	-41	-35	-41	-41	-35	-36	-35	-36	-35	-29	**-40**
V5	-43	13	13	13	13	18	-41	-41	-41	-41	20	14	15	-35	15	15	21	20	21	20	21	27	**0**
V6	-43	-44	-42	-43	-43	-36	-40	-40	-40	-40	-34	-41	-41	-32	-40	-40	-32	-34	-32	-33	-33	-24	**-38**
V7	-10	-10	-9	-9	-10	-10	-9	-9	-9	-8	-4	-9	-9	-4	-9	-8	-8	-9	-8	-9	-4	-3	**-8**
V9	36	-14	-15	37	-14	-7	37	37	39	39	-7	38	38	45	40	40	-5	-4	-5	-4	46	3	**19**
V10	-34	-35	-34	-34	-34	-27	-32	-32	-32	-32	-25	-33	-33	-24	-32	-32	-23	-24	-23	-24	-25	-15	**-30**
V14	-44	-45	-44	-44	-44	-38	-36	-36	-36	-36	-37	-36	-36	-29	-36	-35	-29	-30	-28	-30	-29	-22	**-36**
V15	-48	-48	-47	-48	-48	-44	-38	-38	-38	-38	-43	-38	-38	-34	-38	-37	-33	-34	-33	-34	-34	-30	**-40**
V16	-53	-53	-52	-53	-53	-49	-52	-52	-52	-52	-48	-52	-52	-47	-51	-51	-47	-48	-47	-48	-47	-43	**-50**
V17	-23	-23	-23	-23	-23	-20	-22	-22	-21	-22	-19	-22	-22	-18	-22	-22	-18	-19	-18	-19	-18	-15	**-21**
V18	-20	-19	-16	-18	-43	-10	-14	-14	-42	-41	-6	-14	-14	-5	-40	-40	-4	-33	-4	-32	-5	5	**-21**
V19	-42	-44	-40	-41	-40	-37	-38	-38	-38	-37	-30	-38	-38	-28	-36	-36	-28	-29	-28	-29	-29	-19	**-36**
V21	-20	-19	29	-20	31	-20	32	32	30	30	-9	33	33	-6	30	30	26	29	27	29	-6	-8	**15**
V22	-20	-20	-20	-20	-20	-21	-19	-19	-18	-18	-9	-20	-20	-9	-18	-18	-17	-18	-17	-18	-10	-7	**-18**
V23	-30	-31	-27	-29	-29	-31	-21	-21	-23	-23	-15	-20	-20	-9	-21	-22	-22	-26	-22	-26	-8	-10	**-23**
Total process	-25	-27	-26	-23	-25	-21	-23	-23	-22	-22	-20	-21	-21	-18	-19	-19	-17	-17	-17	-17	-15	-12	**-21**

Figure 12. Effects of joint improvement combinations to reduce iterations of V10 and V22 on individual activities; improvement combinations ordered by increasing expected process duration.

Having conducted this analysis, the design manager has yet to decide which improvement combination to choose for implementation. One of the first considerations to make an educated choice is trading-off effort and duration reductions (Shapiro et al.

2015). For example, the joint improvement combination #1 comes with a greater reduction in duration but smaller reduction in effort compared to #2 (see Figure 11). Also, the suggested improvement combinations are predicted (i.e. theoretical). Each promising combination should be assessed regarding its practical feasibility, as it might not be possible or too expensive in the current situation of the organisation to increase the confidence levels of certain external inputs or to increase the effort investment into certain activities.

The design team decided to consider an improvement combination similar to the joint combination #8 in the next execution of the preliminary fan sub-system preliminary design process. This requires increasing the confidence into the inputs of activities V5, V10 and V22 through providing the responsible designers with standardised guidance and also through additional research, as well as increasing the effort for activities D6, V9 and V21, which is also shown in Figure 12. These changes are expected to reduce both process effort and duration by up to 23% (see combination #8 in Figure 11).

7. Discussion

7.1. Practical usefulness of DPCM

To recapitulate, the DPCM was intended to help design teams (see Sections 2 and 4.4):

(1) Gain understanding of DPC effects on design process performance;

(2) Improve process execution by reacting to and implementing DPCs efficiently;

(3) Improve process planning by prioritising optional DPCs effectively based on costs/benefits.

The identification and comparison of change combinations to reduce iterations presented in the fan sub-system preliminary design process, which led to the suggestion of improvements of the overall process duration and effort of up to 23%, was clearly focused on the third goal, i.e. to improve process planning. However, implicitly this example also verified the method's usefulness to fulfil the first goal, as the effects of various changes on the overall process (see Figure 11) and also on single activities (see Figure 12) were examined. In addition to the presented analysis, which corresponds to analysis 3a in Section 4.4, each of the other suggested analysis types was applied to the fan sub-system preliminary design process, with the following results:

(1) To gain understanding of DPC effects a criticality analysis of one-at-a-time decreases of deliverable confidence and activity effort levels was carried out, leading to the identification of critical inputs and activities, each of which could increase process duration by up to 4% and effort by up to 5% (see Section 4.4, analysis 1);

(2) To improve process execution, efficient reactions to these critical changes were identified, some of which could potentially even overcompensate the negative change effects and improve the overall process performance (see Section 4.4, analysis 2);

(3) In addition to the presented improvement combinations to reduce iterations, process planning was also supported by suggesting effort reductions in activities where additional effort would not result in higher levels of output confidence based on their confidence mappings (see Section 4.4, analysis 3b) – such reductions could improve process duration by 5% and process effort by 7%. Furthermore, improvement combinations to increase the confidence into major process outputs were suggested (see Section 4.4, analysis 3c) and, according to

the simulation results, these may even come for free, i.e. without increasing process duration and effort, as they rely on increased confidence levels in internal deliverables, which as a side-effect reduce iterations in the process.

All of the results summarised above, were discussed with design engineers and managers, who verified their plausibility. Since some of the suggested improvements will be considered in the next execution of the design process (see Section 6.2) the DPCM's practical usefulness can be, at least thus far, confirmed.

7.2. Limitations and future work

Since there are multiple types of DPCs caused by a variety of reasons (see Section 2) and because design processes can differ substantially in terms of their planning and execution practices, it is important to point out the intended scope and potentially limiting assumptions of the DPCM.

First, the DPCM is developed for design processes of artefacts that are not designed from scratch but are modifications of existing artefacts. These so-called evolutionary design processes constitute the majority of product designs (Bucciarelli 1994). This application focus allows the collection of data and prediction of DPC effects with reasonable confidence, which would be much more difficult with radically new designs where mature process plans are unlikely to exist and processes are less well understood.

Second, the concept's fundamental assumption is that DPCs can be represented as combinations of changes in activity effort and deliverable confidence. Both the examined literature and the industrial study have confirmed the flexibility of this approach to capture many types of DPCs. In fact, the replication of effects of a past change in a major input as well as the examination of a prospective structural change produced plausible results for the examined fan sub-system preliminary design process. Nevertheless, particularly a major structural DPC, for example, an automation of the design process, which affects multiple activities, their inputs, outputs and interdependencies, may be difficult to model in this manner. Not only, would this require the user to adapt the process model manually (see Table 1, requirement 5), including adding, taking out or reconnecting activities and deliverables, but more importantly, it would be difficult for the user to assess the changed confidence into the affected outputs and the corresponding effects on the iterative behaviour of the design process. This challenge to represent effects of major structural changes is exacerbated given the small number of effort and confidence levels considered in the method (see also discussion of the DPCM's implementation below). Therefore, the applicability of the DPCM is currently limited to local changes, which affect only a few activities and/or deliverables. Overall, additional research to examine the representability of empirical changes in different design processes using the DPCM would be beneficial.

Third, the DPCM can only predict the performance of the examined process part. Effects of potential changes or improvements on the sub-processes upstream or downstream from the examined process part are not considered. For example, in the analysis of improvement combinations to reduce iterations (see Section 6.2), the method did not explicitly account for the effort potentially required in upstream processes to increase the confidence into inputs of the examined process. Overall, this modelling assumption is not expected to be critical, as further investigations showed that it does not necessarily require additional effort to increase confidence levels of deliverables (see Section 7.1) and, that once produced, many of these inputs can be utilised repetitively for future design processes. Conversely, downstream processes may benefit from increases in the confidence of output deliverables of the examined process, which is also not explicitly predicted by the DPCM. Thus, an interesting direction for future research is to examine the total costs/benefits of local changes and improvements for the overall design process.

Fourth, with regard to the DPCM's implementation, particularly the choice of a small number of discrete levels to measure confidence and effort and the relatively simple functional formalisation of the fundamental relationships between confidence, effort and iterations should be discussed. The choice of a small number of discrete confidence levels was also made in Clarkson and Hamilton (2000), who reported that designers are more comfortable with the use of discrete levels than with a (cardinal) numeric representation of confidence – an observation which is also supported by Ullman, Herling, and D'Ambrosio (1997). Also, the reasons why other authors used a finer measurement of confidence (e.g., Flanagan, Eckert, and Clarkson 2007) are not applicable to the DPCM as it is based on a relative rather than an absolute understanding of confidence (see Section 4.2): Applying an absolute understanding of confidence, the confidence level into a deliverable increases after every further transformation through an additional activity. Contrary, applying a relative understanding of confidence, the confidence level into a deliverable does not necessarily increase after every transformation through an additional activity, as it is compared to the usual confidence into this deliverable produced by a comparable activity in similar, past design projects and at an equivalent design stage. Furthermore, the choice of few discrete confidence and effort levels is also reasonable, given the simple functional formalisation of the method's fundamental relationships (see Section 5.2), as a finer measurement brings no advantage if these functions are not designed to operate accordingly. However, as these functions can only be specified based on the experience of designers the authors argue that they should be kept as simple as possible to avoid the impression of a false accuracy and also to support the method's traceability. Thus, these relationships are currently represented through simple mappings of confidence and effort levels to certain outcomes. Nevertheless, some benefits of a finer measurement of confidence and effort together with a more sophisticated formalisation of relationships are conceivable. In fact, this could allow representing the impacts of more moderate or even major (structural) changes on design process performance. In this context, future research should therefore examine whether potential benefits of a finer parametrisation of the method would justify the correspondingly increased data input requirements and the increased computational effort for simulation experiments.

Lastly, it should be noted that the organisational impact of adapting the DPCM has yet to be studied. In particular, future work should explore the requirements and implications of integrating the DPCM concept into the existing design process governance and IT infrastructure of industrial companies.

8. Conclusion

So far research on changes in design has focused on engineering changes, i.e. changes in the product domain. However, the design process, which creates the product and is characterised by the coordinated execution of activities with complex interdependencies, is also subject to change. Such design process changes (DPCs) may comprise various perturbations that affect design activities, their deliverables or process structure, and ultimately impact process performance. As there is still a lack of comprehensive methods to support modelling and analysis of DPCs this article systematically develops such a method, the Design Process Change Method (DPCM).

The key idea of the DPCM is representing DPCs as combinations of changes in activity effort and deliverable confidence, which influence the confidence into downstream deliverables and ultimately impact the iterative behaviour of the design process. This idea is supported by both existing literature and industrial practice and results in a propagation network, which enables analysis of DPC impacts on the holistic process and sub-process level.

The contributions of this research are threefold: first, it systematically develops a concept for the DPCM together with a set of useful analyses based on requirements derived from literature and industrial practice; second, it details and implements the concept computationally using the Applied Signposting Model framework; and third, it demonstrates how to apply the DPCM to improve the understanding of DPC effects on process performance and to support process planning and execution through identifying and prioritising the ‘right’ DPCs and appropriate DPC reactions. An application of the DPCM to the fan sub-system preliminary design process of Rolls-Royce PLC resulted in the suggestion of process improvements which are expected to decrease process duration and effort by up to 23%.

Whilst the DPCM proved useful in its industrial application, there are still some interesting directions for future research: further empirical analyses of changes in industrial design processes to assess their representability using the DPCM; an examination of whether a finer measurement of confidence and effort together with a more sophisticated formalisation of relationships would be beneficial for analysing complex DPCs; an investigation as to the total costs/benefits of changes and improvements in a sub-process on the whole process; and an examination of the implications of integrating the DPCM concept into the existing design process governance and IT infrastructure of industrial companies.

Acknowledgement

The authors would like to thank all Rolls-Royce PLC employees, who contributed to the case study. The authors are also grateful for the valuable feedback of Dr Michael Moss and the anonymous reviewers. This work was funded by Innovate UK/Technology Strategy Board and Rolls-Royce PLC.

References

Ahmad, N., D.C. Wynn, and P.J. Clarkson. 2013. 'Change Impact on a Product and Its Redesign Process: A Tool for Knowledge Capture and Reuse'. *Research in Engineering Design* 24 (3): 219–44.

Browning, T.R., and S.D. Eppinger. 2002. 'Modeling Impacts of Process Architecture on Cost and Schedule Risk in Product Development'. *IEEE Transactions on Engineering Management* 49 (4): 428–42.

Browning, T.R., E. Fricke, and H. Negele. 2006. 'Key Concepts in Modeling Product Development Processes'. *Systems Engineering* 9 (2): 104–28.

Bucciarelli, L.L. 1994. 'Designing Engineers'. Cambridge, MA, USA: MIT Press.

Chalupnik, M.J., D.C. Wynn, C.M. Eckert, and P.J. Clarkson. 2007. 'Understanding Design Process Robustness: A Modelling Approach'. In *16th International Conference on Engineering Design, ICED 2007*, 455–56. Paris, France.

Chua, D.K.H., and M.A. Hossain. 2012. 'Predicting Change Propagation and Impact on Design Schedule due to External Changes'. *IEEE Transactions on Engineering Management* 59 (3): 483–93.

Clarkson, P.J., and J.R. Hamilton. 2000. '"Signposting", a Parameter-Driven Task-Based Model of the Design Process'. *Research in Engineering Design* 12 (1): 18–38.

Cronemyr, P., A. Öhrwall Rönnbäck, and S.D. Eppinger. 2001. 'A Decision Support Tool for Predicting the Impact of Development Process Improvements'. *Journal of Engineering Design* 12 (3): 177–99.

Cross, N., and N.F.M. Roozenburg. 1992. 'Modelling the Design Process in Engineering and in Architecture'. *Journal of Engineering Design* 3 (4): 325–37.

Dvir, D., and T. Lechler. 2004. 'Plans Are Nothing, Changing Plans Is Everything: The Impact of Changes on Project Success'. *Research Policy* 33 (1): 1–15.

Eppinger, S.D. 1991. 'Model-Based Approaches to Managing Concurrent Engineering'. *Journal of Engineering Design* 2 (4): 283–90.

Flanagan T.L., C.M. Eckert, and P.J. Clarkson. 2007. 'Externalizing tacit overview knowledge: A model-based approach to supporting design teams'. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 21(3): 227–242.

Hamraz, B. 2013. 'Engineering change modelling using a function-behaviour-structure scheme'. PhD thesis. University of Cambridge.

Hamraz, B., N.H.M. Caldwell, and P.J. Clarkson. 2013. 'A Holistic Categorisation Framework for Literature on Engineering Change Management'. *Systems Engineering* 16 (4): 473–505.

Hamraz, B., N.H.M. Caldwell, D.C. Wynn, and P.J. Clarkson. 2013. ‘Requirements-Based Development of an Improved Engineering Change Management Method’. *Journal of Engineering Design* 24 (11): 765–93.
Karniel, A., and Y. Reich. 2013. ‘Multi-Level Modelling and Simulation of New Product Development Processes’. *Journal of Engineering Design* 24 (3): 185–210.
Khoo, L.P., C.-H. Chen, and L. Jiao. 2003. ‘A Dynamic Fuzzy Decision Support Scheme for Concurrent Design Planning’. *Concurrent Engineering Research and Applications* 11 (4): 279–88.
Kloss-Grote, B., and M.A. Moss. 2008. ‘How to Measure the Effectiveness of Risk Management in Engineering Design Projects? Presentation of RMPASS: A New Method for Assessing Risk Management Performance and the Impact of Knowledge Management—including a Few Results’. *Research in Engineering Design* 19 (2-3): 71–100.
Lévárdy, V., and T.R. Browning. 2009. ‘An Adaptive Process Model to Support Product Development Project Management’. *IEEE Transactions on Engineering Management* 56 (4): 600–620.
Lévárdy, V., M. Hoppe, S. Wenzel, and A. Vollerthun. 2003. ‘Process Modeling Procedure for Verification, Validation and Testing Planning’. In *Proceedings of the ASME 2003 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Chicago, USA.
Li, W., and Y.B. Moon. 2012. ‘Modeling and Managing Engineering Changes in a Complex Product Development Process’. *The International Journal of Advanced Manufacturing Technology* 63 (9-12): 863–74.
Lukas M, Gärtner T, Rohleder N, Schlick CM (2007) A simulation model to predict impacts of alterations in development processes. In: 9th International Design Structure Matrix Conference, DSM’07. Munich, Germany, pp 127–136.
MacCallum, K.J., and A. Duffy. 1987. ‘An Expert System for Preliminary Numerical Design Modelling’. *Design Studies* 8 (4): 231–37.
Ouertani, M.Z. 2008. ‘Supporting Conflict Management in Collaborative Design: An Approach to Assess Engineering Change Impacts’. *Computers in Industry* 59 (9): 882–93.
Sanchez, S.M., and H. Wan. 2012. ‘Work Smarter, Not Harder: A Tutorial on Designing and Conducting Simulation Experiments’. In *Simulation Conference (WSC), Proceedings of the 2012 Winter*, 1–15.
Shapiro, D., B. Hamraz, A.F. Sommer, and P.J. Clarkson. 2015. ‘Investigating the Impact of Changes in Iteration-Likelihoods on Design Process Performance’. *Concurrent Engineering: Research and Applications* 23 (1): 1–15.
Shapiro, D., A.F. Sommer, and P.J. Clarkson. 2015. ‘An Investigation of Design Process Changes’. In *Proceedings of the 20th International Conference on Engineering Design (ICED 15)*, 451–60. Milan, Italy.
Shapiro, D., and P.J. Clarkson. 2016. ‘A Concept for Modelling and Analysing Design Process Changes’. To be published in *Proceedings of Design 2016*. Dubrovnik, Croatia.
Sim, S.K., and A.H.B. Duffy. 2003. ‘Towards an Ontology of Generic Engineering Design Activities’. *Research in Engineering Design* 14 (4): 200–223.
Suss, S., and V. Thomson. 2012. ‘Optimal Design Processes under Uncertainty and Reciprocal Dependency’. *Journal of Engineering Design* 23 (10-11): 826–48.

Ullman, D.G., D. Herling, and B. D'Ambrosio. 1997. 'What to Do next: Using Problem Status to Determine the Course of Action'. *Research in Engineering Design* 9 (4): 214–27.
Vieira, H., S.M. Sanchez, K.H. Kienitz, and M.C.N. Belderrain. 2011. 'Improved Efficient, Nearly Orthogonal, Nearly Balanced Mixed Designs'. In *Proceedings of the 2011 Winter Simulation Conference (WSC)*, 3600–3611.
Wynn, D.C., C.M. Eckert, and P.J. Clarkson. 2006. 'Applied Signposting: A Modeling Framework to Support Design Process Improvement'. In *Proceedings of the ASME International Design Engineering Technical Conferences, DETC2006-99402*, 1–10. Philadelphia, PA, US.
Wynn, D.C., K. Grebici, and P.J. Clarkson. 2011. 'Modelling the Evolution of Uncertainty Levels during Design'. *International Journal on Interactive Design and Manufacturing* 5 (3): 187–202.
Wynn, D.C., N.H.M. Caldwell, and P.J. Clarkson. 2014. 'Predicting Change Propagation in Complex Design Workflows'. *Journal of Mechanical Design* 136 (8): 081009–1 – 081009–13.

Tables

Table 1: Conceptual ideas to fulfil the identified functional method requirements.

Functional requirement	Conceptual idea to fulfil the requirement
1. Activity-based modelling of evolutionary design processes (Khoo, Chen, and Jiao 2003; Wynn, Caldwell, and Clarkson 2014)	Theoretically, the DPCM could be based on various activity-network-based frameworks. The ASM (Wynn, Eckert, and Clarkson 2006) is suggested because it allows capturing complex interrelations between activities and deliverables, which is key for the analysis of DPCs. Also, it was specifically developed for modelling design processes and thus, contains many design-focused features so that it is convenient to use.
2. Modelling iteration (Chalupnik et al. 2007; Li and Moon 2012)	As DPCs can affect the level of uncertainty in the process and may trigger iterations, which substantially impact process performance (Eppinger 1991), it is suggested to model the occurrence of iterations dependent on uncertainty (see, e.g., Lévárdy and Browning 2009).
3. Modelling changes in activities (Cronemyr, Öhrwall Rönnbäck, and Eppinger 2001; Khoo, Chen, and Jiao 2003)	It is suggested to represent changes in activities as changes in the associated effort (Lukas et al. 2007), which implicitly considers both changes in activity durations and in their resource requirements, and increases the flexibility of modelling real-world processes.
4. Modelling changes in deliverables (Chua and Hossain 2012; Wynn, Caldwell, and Clarkson 2014)	Based on the existing literature (see, e.g., Chua and Hossain 2012), it is suggested to account for specific changes in deliverables by capturing their potential of causing iterations.
5. Modelling structural changes (Karniel and Reich 2013)	It is suggested to adapt process plans manually in order to represent structural changes because rule-based automatic adaptation of plans adds significant complexity to the method and also does not work for every change case (Karniel and Reich 2013). The DPCM's user will thus decide, e.g., whether a new activity can be added without adding new deliverables.
6. Modelling propagating DPC effects (Ahmad, Wynn, and Clarkson 2013; Ouertani 2008)	It is suggested to consider DPC propagation between activities and deliverables (Wynn, Caldwell, and Clarkson 2014) in order to equally capture the lower-level effects of changes in activities, deliverables and structural changes, i.e. activity-deliverable relationships. Moreover, in order to limit the complexity of modelling, the method concept is restricted to the process domain so that change propagation to elements of the product domain, such as components, is not considered.

7. Identifying critical DPCs, reactions and DPC-based process improvements (Chalupnik et al. 2007; Cronemyr, Öhrwall Rönnbäck, and Eppinger 2001)	To increase the practical usefulness the identification and comparison of multiple alternative candidates for critical DPCs, reactions and process improvements (Browning and Eppinger 2002), rather than the identification of a single theoretical worst or best case, should be supported.
8. Analysing DPC impacts (Chua and Hossain 2012; Lukas et al. 2007)	Process simulations are suggested to assess DPC impacts because closed-form analysis is often not possible for complex, stochastic networks (Shapiro et al. 2015).

List of Figures