INTERNATIONAL DESIGN CONFERENCE - DESIGN 2016
Dubrovnik - Croatia, May 16 - 19, 2016.

A CONCEPT FOR MODELLING AND ANALYSING DESIGN PROCESS CHANGES

D. Shapiro and P. J. Clarkson

Keywords: design process, process modelling, process analysis, process change, iteration

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 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, e.g., 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 because inputs of activities change [Chua and Hossain 2012]. Such design process changes (DPCs) can propagate, affecting key process performance metrics. In particular, it can be difficult to predict their impacts on process duration and cost [Shapiro et al. 2015b], which may ultimately also affect customer satisfaction. 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 26 existing methods that account for DPCs [Shapiro et al. 2015b]. Since 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.

This article describes the systematic development of a concept for such a support method and is organised as follows: Section 2 provides an overview on DPCs and existing support methods. Section 3 explains the research method. Section 4 derives requirements for a comprehensive support method based on the literature discussed in Section 2. Building on these requirements, Section 5 develops a broad method concept, defines and links the method's fundamental elements, and finally examines a set of process analyses, which are supported by the method. Section 6 discusses the key assumptions and the scope of the derived method concept. Section 7 summarises and concludes the article.

2. Literature review

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 of 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 et al. 2015b, p. 453].

This definition emphasises three fundamental types of DPCs, 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 their durations, 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 sequence of activity execution. Structural changes immediately affect at least one activity and one deliverable in the process. If, e.g., a new activity is added to the process, it might require existing deliverables as inputs and will necessarily produce a new output or change an existing output.

The current DPC literature mostly discusses DPCs that originate from ECs [Chua and Hossain 2012], [Li and Moon 2012] 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 may 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 for DPCs, which do not necessarily originate from the product. For example, DPCs may come from process improvements suggested by new project-team members, or originate from a lack of competence of designers 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 [Lukas et al. 2007] 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 DPC propagation make it difficult to predict DPC effects on process performance.

As the few existing design process change management (DPCM) methods are often inspired by EC-propagation methods in the product domain, they have analogous goals [Hamraz et al. 2013a], [Shapiro et al. 2015b], i.e. to support design teams to

1. Gain understanding of impacts and process-performance risks stemming from DPCs;
2. Improve process execution by reacting adequately and implementing DPCs efficiently;
3. Improve process planning by prioritising optional, positive DPCs effectively.

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 et al. 2015b] identified only 26 methods among which 17 solely examine effects of deliverable changes e.g. [Ouertani 2008], [Ahmad et al. 2013]. Among the other nine methods there is none which covers investigations of all three major DPC types. Also, the reviewed DPCM methods offer highly varying features: Some methods, e.g., can help identifying activities affected by a DPC [Ahmad et al. 2013], while other methods additionally suggest an activity sequence for DPC implementation [Khoo et al.

2003]. Or, some methods explicitly consider DPC implementation timing [Chua and Hossain 2012], while others assume stochastic DPC arrival rates [Li and Moon 2012]. One common feature among most methods (although treated in different ways) 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 DPCM method [Shapiro et al. 2015b].

3. Research method

To address the previously identified need of developing a comprehensive DPCM method, this research follows a systematic procedure analogous to an actual design process, i.e. deriving requirements, developing alternative concepts, selecting and elaborating a concept, detailing and computationally implementing the method as well as evaluating and refining the method. This article describes the method's conceptual design. Its detail design and evaluation are subject of the authors' ongoing research.
The research started with a requirements analysis to define the specific needs that the intended DPCM method should fulfil (Section 4). 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 intended DPCM method. The authors then detailed the concept's fundamental elements and their interrelations (Sections 5.3 and 5.4) and defined a set of analyses to improve process understanding, execution and planning, which will be enabled through such a support method (Section 5.5).
Throughout this article the rationale for the choices of requirements, alternative concept ideas, a single broad concept as well as its elements is based on three sources, as discussed in the respective sections: first, the authors' literature survey on DPCs [Shapiro et al. 2015b] and other key engineering design literature; second, the authors' investigation of changes in iteration-likelihoods [Shapiro et al. 2015a]; third, an ongoing industrial study at a leading aerospace company.

4. Requirements for a method to support modelling and analysis of DPCs

Although the goals of the intended DPCM method are known (Section 2), its design, like every systematic design of products and systems, should start with a requirements analysis to define the specific needs and conditions that it needs to fulfil. This section describes the procedure of deriving requirements for such a method and summarises its results (Table 1).

Table 1. Functional requirements for a comprehensive DPCM method

Functional requirement	Description	Selected sources for rationale
1. Activity-based design processes modelling	Model evolutionary design processes of different granularities and complexities as activity-based networks.	[Khoo et al. 2003], [Karniel and Reich 2013]
2. Modelling uncertainty	Model uncertain activity outcomes, e.g. iteration, uncertain durations and uncertain resource requirements.	[Li and Moon 2012], [Shapiro et al. 2015a]
3. Modelling changes in activities	Model changes in activity attributes, such as activity durations and resource requirements.	[Khoo et al. 2003], [Lukas et al. 2007]
4. Modelling changes in deliverables	Model changes in deliverables, such as enhancements, corrections and scope changes.	[Ouertani 2008], [Chua and Hossain 2012]
5. Modelling structural changes	Model structural changes, such as adding or removing activities and/or deliverables and changes in process execution logic.	[Karniel and Reich 2013]
6. Modelling DPC effects	Model (in-)direct DPC effects accounting for the interconnectivity of activities and deliverables in design processes.	[Ahmad et al. 2013], [Hamraz et al. 2013b]
7. Identifying critical DPCs, reactions and process improvements	Provide analyses to systematically identify potentially critical DPCs, adequate reactions, and improvements through positive DPCs.	[Chalupnik et al. 2007], [Shapiro et al. 2015a]

8. Analysing DPC impacts	Provide analyses to assess the impacts of DPCs, of adequate reactions, and of improvements through positive DPCs on process performance.	[Chua and Hossain 2012], [Hamraz et al. 2013b]

Given the few existing DPCM methods it is not surprising that no literature was found covering an explicit derivation of requirements for such a method. However, the publications examined in Section 2 usually discuss selected requirements or consider these implicitly in their respective methods. Therefore, studying these publications carefully, key features of the described methods were extracted to produce an exhaustive requirements list for a DPCM method. Additionally, a few requirements were derived from the authors' investigation of iteration-likelihood changes [Shapiro et al. 2015a]. Finally, some requirements were added from the very comprehensive, requirements-based development of an ECM tool by Hamraz et al. [2013b] and transferred from the product- into the process-domain. The resulting requirements were then checked for practical relevance based on an ongoing industrial study.
This approach resulted in the identification of 18 requirements for a DPCM method, including functional requirements and general requirements, the latter of which describe the desired conditions for method inputs, application and outputs. For the aim of this paper, i.e. to systematically derive a comprehensive DPCM method concept, the functional requirements are of predominant importance and thus, are described in Table 1 (please contact the first author for a full list of requirements). The table also provides sources for the selection rationale of each functional requirement.

5. Conceptual design of a method to support modelling and analysis of DPCs

5.1 Comparison of alternative method concepts and choice of most suitable concept idea

This section compares different options to fulfil the derived functional requirements for a DPCM method using a morphological chart (Table 2). Most of the options are inspired by existing design process modelling literature.

Table 2. Morphological chart to derive alternative DPCM method concepts

Functional requirement	Concept options					
1. Activity-based design processes modelling	CPM/ PERT [PMI 2013]	DSM [Steward 1981]	Signposting [Clarkson and Hamilton 2000]	ASM [Wynn et al. 2006]	Petri Nets [Petri 1962]	IDEF0 [NIST 1993]
2. Modelling uncertainty	Deterministic occurrence of iterations	Probabilistic occurrence of iterations through fixed iteration-likelihoods	Probabilistic occurrence of iterations driven by uncertainty in the design process			
3. Modelling changes in activities	Representation of changes in deterministic activity attributes	Stochastic representation of activity attributes, i.e. implicit modelling	Representation of changes in stochastic activity attributes			
4. Modelling changes in deliverables	Qualitative representation of specific changes	Stochastic occurrence of likely changes, i.e. implicit modelling	Quantitative representation of specific changes			
5. Modelling structural changes	Qualitative (visual) representation and manual adaptation of process flowcharts	Qualitative and quantitative representation and manual adaptation of process flowcharts	Qualitative and quantitative representation and automated generation of adapted process flowcharts			
6. Modelling DPC effects	Consideration of DPC effects on holistic process performance	Consideration of effects on sub-process performance	Consideration of DPC effects on both sub-process and holistic process performance			

7. Identifying critical DPCs, reactions and process improvements	Qualitative description of alternative candidates for critical DPCs, reactions and process improvements	Quantitative process optimisation	Quantitative heuristic to identify single candidate	Quantitative heuristic to identify alternative candidates, e.g., through design of experiments
8. Analysing DPC impacts	Qualitative process analysis	Quantitative closed-form process analysis	Quantitative process simulations	

The selected concept option for each of the eight functional requirements is highlighted through thick borders and its respective selection rationale is discussed as follows:

1. Although the method could be built on any of the listed activity-network based frameworks, the Applied Signposting Model (ASM) [Wynn et al. 2006] is chosen particularly because of two reasons: First, it allows capturing complex interrelations between activities and deliverables, which is key for the analysis of DPCs. Second, as it was specifically developed for the purpose of modelling complex design processes, it contains many useful design-focused features (e.g., built-in representation of iteration) and is thus, very convenient to use.
2. As DPCs cause iteration (Section 2), which is a major driver of process performance, this aspect should be captured thoroughly in the method. Also, the authors' prior research showed that fixed likelihoods may be inadequate to represent iteration [Shapiro et al. 2015a]. Thus, a probabilistic occurrence of iteration driven by uncertainty e.g. [Lévárdy and Browning 2009] is suggested.
3. Representing changes in stochastic activity attributes e.g. [Chalupnik et al. 2007] is the most flexible concept option, which proved relevant in an ongoing industrial study.
4. This industrial study also supports the findings from literature (Section 2) that changes in deliverables may trigger iterations in evaluation activities causing substantial rework. Hence, it is suggested to consider specific deliverable changes explicitly e.g. [Chua and Hossain 2012].
5. As automatically adapting process flowcharts after structural changes adds significant complexity to the method and also does not work in every case [Karniel and Reich 2013], a qualitative and quantitative representation of structural DPCs with a manual adaptation of flowcharts is chosen. The envisioned method's user will thus, decide such things as whether a new activity can be added without adding new deliverables and reworking other activities.
6. To increase managerial insights and practical usefulness (e.g., for making resourcing decisions) the method should not only consider change effects on the holistic process but also on the activity/sub-process level e.g. [Shapiro et al. 2015a].
7. For the same reasons, i.e. to increase managerial insights and usefulness, the identification of multiple alternative candidates for critical DPCs, adequate reactions and process improvements, rather than the identification of a single theoretical worst or best case, should be supported.
8. Lastly, process simulations are selected as a means to assess DPC impacts because closed-form analysis is often not possible for complex, stochastic networks [Shapiro et al. 2015a] and also, additional insights compared to a purely qualitative analysis may be gained.

5.2 Fundamental method

This section provides an overview of the overall concept suggested for the novel DPCM method (Figure 1), which integrates the selected concept options from the morphological chart.

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 (Table 2, requirements 3-5). In addition to these direct effects on singular process elements, there are indirect effects on its iterative behaviour, as it is driven by confidence (or uncertainty, Table 2, requirement 2). These indirect effects 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 design 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.

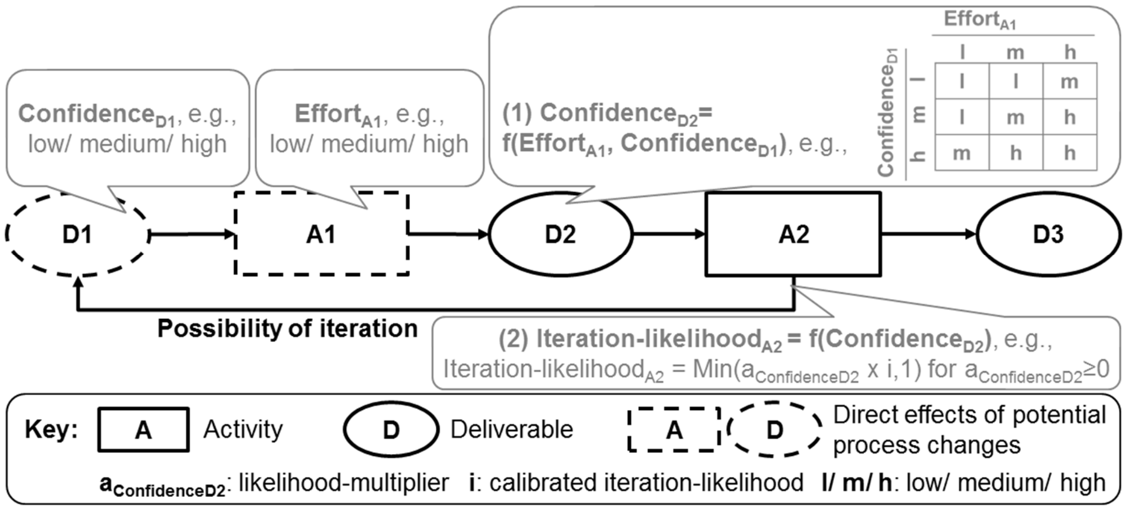

Figure 1. Overall concept for the suggested DPCM method and initial specification of method elements and relationships

This concept can be implemented in ASM, which also supports quantitative process analyses through Monte-Carlo simulations (Table 2, requirements 1 and 8). Moreover, ASM allows the definition of process variables so that change effects can be captured for sub-processes as well as the holistic process (Table 2, requirement 6). Finally, the propagation network described above can be examined to suggest alternative candidates for critical DPCs, reactions and process improvements (Table 2, requirement 7).

5.3 Fundamental method elements

The concept is based on two fundamental elements: activities and deliverables (Figure 1).

Activities are constituent parts of a process, i.e. sub-processes, that 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 (Figure 1), 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. Due to this broad definition, effort will need to be approximated in the implementation of the suggested DPCM method concept, e.g., through the utilisation of human resources over the durations of activities, so that it could be measured in person-days. Besides, based on the examined generic activity formalism, all design activities may trigger iterations, as they are executed with imperfect knowledge.

Given that the concept is based on ASM, it assumes that the design can be represented as the generation and refinement of deliverables through activities (Figure 1). Such a deliverable can be quantitative or qualitative and describe any characteristic of the product, e.g., 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 et al. 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, p. 24] 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 into 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 (Although the goals of the intended DPCM method are known (Section 2), its design, like every systematic design of products and systems, should start with a requirements analysis to define the specific needs and conditions that it needs to fulfil. This section describes the procedure of deriving requirements for such a method and summarises its results (Table 1).
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 may then impact the iterative behaviour downstream.

5.4 Fundamental relationships between method elements

The first fundamental relationship is a dependency between the confidence into an activity's inputs, the effort invested into an activity and the confidence into the activity's resulting output (Figure 1), i.e.

$$Confidence_{Output} = f\left(Effort, Confidence_{Inputs}\right). \tag{1}$$

This relationship is discussed in the engineering design literature from various perspectives e.g. [MacCallum and Duffy 1987], [Wynn et al. 2011]. Intuitively it is positive, i.e. the confidence into a design activity's outputs 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.
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 confidence into outputs 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 relationship. Also, as outputs can depend on multiple inputs, the implementation of the proposed concept needs to include means to aggregate inputs with different levels of confidence into a single measure of input confidence per activity, e.g., by assigning a weight to each input depending on its relevance for the output.
Furthermore, it is noteworthy that an additional network of interactions is introduced to the ASM framework through the first 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.
The second fundamental relationship underlying the proposed method concept is a dependency between the confidence into an activity's inputs and its likelihood to trigger iterations (Figure 1), i.e.

$$Iteration - likelihood = f\left(Confidence_{Inputs}\right). \tag{2}$$

This relationship has also been discussed by multiple authors in the engineering design literature e.g. [Lévárdy and Browning 2009], [Wynn et al. 2011]. Intuitively it is negative, i.e. iteration-likelihoods tend to decrease for high confidence levels in input deliverables. 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 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 still 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 activity, which is iterated 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 should be emphasised that this relationship can be only implemented for explicitly captured possibilities of iteration, which are represented through feedback-loops (Figure 1). The modeller thus, needs to carefully choose the model's level of granularity so that relevant iterations are not obscured.

Overall, the two relationships have an important interplay: If, e.g., a DPC led to insufficient effort devoted to the process upstream, this could result in a lower output confidence, which would increase the likelihood of triggering iterations downstream, potentially increasing the overall process duration and effort. Such iterations would also increase the upstream effort so that output confidence would increase and the likelihood of further iterations would decrease. Thus, in the proposed concept iterations function as a control mechanism for the confidence into the design – an intuitive and intended effect.

Initial definitions of method elements and relationships, which can be specified in practice, have already been developed for the suggested concept and have been also computationally implemented in a prototype method. For this, a discrete confidence level, i.e. low/ medium/ high, needs to be specified for every independent input deliverable of the examined process by the responsible design team (Figure 1). Similarly, boundaries need to be defined for each activity to assign a low, medium or high activity effort level during simulations, as activity effort is calculated based on specified probability distributions and accumulates due to rework. The confidence levels of intermediate deliverables are then calculated based on their specified confidence mappings, implementing the first fundamental relationship (Figure 1). To implement the second fundamental relationship, the calibrated iteration-likelihoods of iterative activities are adapted by multipliers, which are quantified depending on the level of input confidence, so that the originally specified iteration-likelihoods are replicated on average during simulations (Figure 1).

5.5 Possibilities of method application

To recapitulate, the proposed method is intended to help design teams gain understanding and improve process execution and planning based on the examination of DPC effects (Sections 2 and 4). Accordingly, the method should offer a tool box with analysis possibilities, which address each of these points. The following analysis tool box (Figure 2), which is based on the method elements and relationships established in the previous sections, is envisioned as part of the method:

Analyses to gain understanding comprise the preventive identification of potentially critical DPCs by examining the impacts of decreasing the confidence into inputs (1a in Figure 2) or of reducing activity effort (1b in Figure 2) on process duration and effort. These analyses can be also extended to actual DPCs, which could comprise a mixture of concurrent input confidence and activity effort changes.

Analyses to improve execution encompass the examination of mitigating reactions to reductions in input deliverable confidence (2a in Figure 2) and/ or activity effort (2b in Figure 2) through investing more effort in downstream activities so that an increase of process duration and effort is minimised or avoided.

Lastly, three analysis types to improve planning, based on identifying and prioritising optional, positive DPCs, are suggested. The first type examines combinations of confidence increases in specific process inputs and effort investments in specific activities to reduce unwanted iterations and overall process duration and effort (3a in Figure 2). The second type identifies activities, which should be executed as lean as possible as additional effort does not increase their output confidence based on the specifications of the first fundamental relationship (3b in Figure 2), again to reduce overall process duration and effort.

The third type assesses combinations of confidence increases in specific inputs and effort investments into specific activities to increase the confidence into major process outputs (3c in Figure 2).
Overall, the suggested method is thus, envisioned to be used for the investigation of various ‘what if?’ scenarios during design process planning and execution.

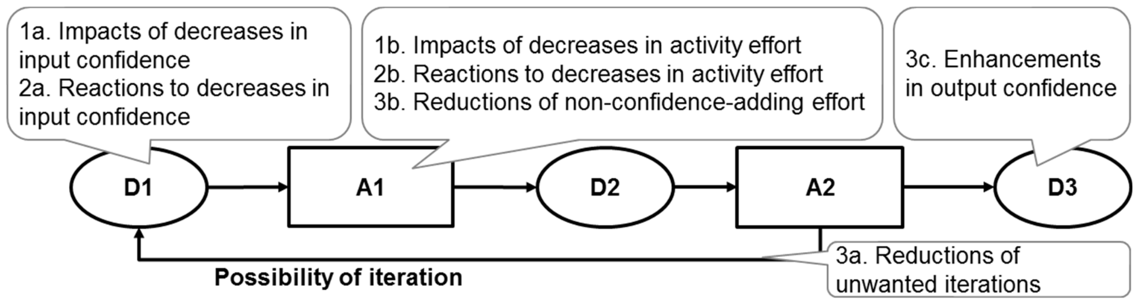

Figure 2. Types of analysis supported by the method

6. Discussion

This section first examines the suggested concept’s intended scope and major assumptions before it discusses the current state of the authors’ research of advancing this concept into a usable method.
As there are multiple types of DPCs caused by a variety of reasons (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 envisioned DPCM method.
First, this method concept is developed for design processes of artefacts that are not designed from scratch but are modifications of existing artefacts. Such evolutionary design processes constitute the majority of product designs [Bucciarelli 1994]. This scope allows predicting DPC effects with reasonable confidence, which is not possible in radically new designs without reliable process plans.
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 authors’ ongoing industrial study have confirmed the flexibility of this approach to capture many types of DPCs. Nevertheless, particularly major structural DPCs, e.g., an automation of design activities, which affects their inputs, outputs and execution sequence, may be difficult to model in this manner and require a substantial manual adaptation of the process model by the user (Section 5.1).
Lastly, although capturing the influence of uncertainty on process performance is a key feature of the suggested DPCM method, the method is not conceptualised to determine the effort required in activities to deal with different levels of uncertainty e.g. [Wynn et al. 2011], but rather treats effort as an independent variable that affects output confidence. This is particularly due to the relative understanding of uncertainty (i.e., confidence, Section 5.3) applied in this research and the authors’ experience with evolutionary design processes where utilised methods are well defined and unlikely to change depending on the level of relative input confidence. Also, other factors than uncertainty can be conceived to significantly impact activity effort: particularly available resources, budget and time.
A method prototype (Section 5.4) has been applied in the authors’ ongoing industrial study, following the analyses described in Section 5.5, and led to promising results, which were verified by the design team. The authors plan to discuss the method’s detail design and application in future publications.

7. 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 a concept for such a method and suggests various applications.

The contributions of this article are twofold: First, it derives and examines a set of functional requirements for a comprehensive design process change management (DPCM) method based on existing literature. Second, it provides a working basis for the detail design and implementation of such a DPCM method by selecting a feasible concept, which fulfils the derived requirements, and studying its fundamental elements, their relationships and potential applications.

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

The authors' current research is focusing on evaluating and refining the DPCM method, which they have built based on the suggested concept, through its application to further industrial cases. Once fully developed, the method will enable design teams to conduct various useful analyses, which will enhance the understanding of the impacts of DPCs on process performance, improve process execution through suggesting reactions to DPCs, and improve process planning through identifying and prioritising optional, positive DPCs. Future publications on the method's detail design and application are planned.

Acknowledgement

This work was funded by Innovate UK/Technology Strategy Board and Rolls-Royce Plc.

References

Ahmad, N., Wynn, D. C., Clarkson, P. J., "Change impact on a product and its redesign process: a tool for knowledge capture and reuse", Research in Engineering Design, Vol. 24, No. 3, 2013, pp. 219–244.

Browning, T. R., Fricke, E., Negele, H., "Key concepts in modeling product development processes", Systems Engineering, Vol. 9, No. 2, 2006, pp. 104–128.

Bucciarelli, L. L., "Designing engineers", MIT Press, Cambridge, MA, USA, 1994.

Chalupnik, M. J., Wynn, D. C., Eckert, C. M., Clarkson, P. J., "Understanding design process robustness: A modelling approach", Proc. of the 16th Int. Conf. on Eng. Design, ICED 2007, Paris, France, 2007, pp. 455–456.

Chua, D. K. H., Hossain, M. A., "Predicting change propagation and impact on design schedule due to external changes", IEEE Transactions on Engineering Management, Vol. 59, No. 3, 2012, pp. 483–493.

Clarkson, P. J., Hamilton, J. R., "'Signposting', a parameter-driven task-based model of the design process", Research in Engineering Design, Vol. 12, No. 1, 2000, pp. 18–38.

Dvir, D., Lechler, T., "Plans are nothing, changing plans is everything: The impact of changes on project success", Research Policy, Vol. 33, No. 1, 2004, pp. 1–15.

Hamraz, B., Caldwell, N. H. M., Clarkson, P. J., "A holistic categorisation framework for literature on engineering change management", Systems Engineering, Vol. 16, No. 4, 2013, pp. 473–505.

Hamraz, B., Caldwell, N. H. M., Wynn, D. C., Clarkson, P. J., "Requirements-based development of an improved engineering change management method", Journal of Eng. Design, Vol. 24, No. 11, 2013, pp. 765–793.

Karniel, A., Reich, Y., "Multi-level modelling and simulation of new product development processes", Journal of Engineering Design, Vol. 24, No. 3, 2013, pp. 185–210.

Khoo, L. P., Chen, C.-H., Jiao, L., "A dynamic fuzzy decision support scheme for concurrent design planning", Concurrent Engineering Research and Applications, Vol. 11, No. 4, 2003, pp. 279–288.

Lévárdy, V., Browning, T. R., "An adaptive process model to support product development project management", IEEE Transactions on Engineering Management, Vol. 56, No. 4, 2009, pp. 600–620.

Li, W., Moon, Y. B., "Modeling and managing engineering changes in a complex product development process", The International Journal of Advanced Manufacturing Technology, Vol. 63, No. 9-12, 2012, pp. 863–874.

Lukas, M., Gärtner, T., Rohleder, N., Schlick, C. M., "A simulation model to predict impacts of alterations in development processes", Proc. of the 9th Int. DSM Conf., DSM'07, Munich, Germany, 2007, pp. 127–136.

MacCallum, K. J., Duffy, A., "An expert system for preliminary numerical design modelling", Design Studies, Vol. 8, No. 4, 1987, pp. 231–237.

NIST., "National Technical Information Service, U.S. Department of Commerce, Federal Information Processing Standards Publication 183 (FIPSPUB 183), Integration definition for function modeling (IDEF0)", 1993.

Ouertani, M. Z., "Supporting conflict management in collaborative design: An approach to assess engineering change impacts", Computers in Industry, Vol. 59, No. 9, 2008, pp. 882–893.

Petri, C. A., "Fundamentals of a theory of asynchronous information flow", Proceedings of IFIP 1962, Amsterdam, Netherlands, 1962, pp. 386–390.

PMI, "A guide to the project management body of knowledge: PMBOK Guide", Project Management Institute, Newtown Square, PA, USA, 2013.

Shapiro, D., Hamraz, B., Sommer, A. F., Clarkson, P. J., "Investigating the impact of changes in iteration-likelihoods on design process performance", Concurrent Engineering, Vol. 23, No. 1, 2015a, pp. 1–15.

Shapiro, D., Sommer, A. F., Clarkson, P. J., "An investigation of design process changes", Proceedings of the 20th International Conference on Engineering Design (ICED 15), Milan, Italy, 2015b, pp. 451–460.

Sim, S. K., Duffy, A. H. B., "Towards an ontology of generic engineering design activities", Research in Engineering Design, Vol. 14, No. 4, 2003, pp. 200–223.

Steward, D. V., "The design structure system: A method for managing the design of complex systems", IEEE Transactions on Engineering Management, Vol. 28, No. 3, 1981, pp. 71–74.

Wynn, D. C., Eckert, C. M., Clarkson, P. J., "Applied signposting: a modeling framework to support design process improvement", Proceedings of the ASME International Design Engineering Technical Conferences, DETC2006-99402, Philadelphia, PA, US, 2006, pp. 1–10.

Wynn, D. C., Grebici, K., Clarkson, P. J., "Modelling the evolution of uncertainty levels during design", International Journal on Interactive Design and Manufacturing, Vol. 5, No. 3, 2011, pp. 187–202.

Daniel Shapiro, PhD Student
University of Cambridge, Department of Engineering
Wolfson College, CB39BB Cambridge, United Kingdom
Email: ds678@cam.ac.uk