Research in Engineering Design
https://doi.org/10.1007/s00163-022-00386-z

ORIGINAL PAPER

Feedback systems in the design and development process

David C. Wynn[1] · Anja M. Maier[2,3]

Received: 31 March 2021 / Revised: 7 March 2022 / Accepted: 11 March 2022

Abstract
Feedback is essential in the design and development process, occurring in the generation of new designs, in the adaptation of development projects to emerging information, and in coordination and collaboration of project participants—among many other aspects. Feedback also contributes to development project complexity and may cause resistance to desirable changes. But despite the importance of feedback in the design and development process (DDP), relatively few publications have examined this topic in an integrated way. This article makes two contributions towards addressing the gap. First, a conceptual framework is developed to organise perspectives on feedback in the DDP literature. The framework shows how feedback occurs at different levels of the design and development process and how it affects important DDP behaviours, namely goal-seeking, learning and emergence. Second, a system-theoretic model of feedback situations in the design and development process is introduced to synthesise key ideas. We provide concrete examples to show how this new model can be used to frame DDP situations and draw out feedback-related insight.

Keywords Feedback · Systems · Cybernetics · Design and development process · Engineering design · Feedback system function structure · FS^2 model

1 Introduction

> "The feedback loop [is] the basic element from which systems are assembled" (Forrester 1968).

The design and development process (DDP) occurs in systems of human activity, and as such can only be perceived incompletely from the unique standpoint of each participant or observer. In consequence, the DDP has been analysed from many perspectives. Some perspectives offer immediate guidance for practice, while others are theory-oriented. Theoretical perspectives can require more interpretation to apply, but typically offer insight into a broader range of situations. Their relatively high level of abstraction is also appropriate when a situation is rich in ambiguity and complexity—as is often the case in design and development.

This article offers a system-theoretic perspective on the design and development process in terms of the feedback that interconnects its behaviours and outcomes. We consider two closely related views on feedback:

- Feedback in terms of cybernetic (self)regulation or 'steering'. Here, the term describes situations in which goal-directed action works to reduce the gap between the perceived and desired states of a system;
- Feedback in terms of circular flows of influence. Here, feedback can be perceived wherever the output of a process later influences its input, directly or indirectly. This differs in an important way from the oft-discussed concept of reciprocal dependency, which may not imply a process.

In both these views, the emphasis is placed on how feedback causes a system to evolve over time. For instance, feedback is often associated with control and stabilisation, or with virtuous and vicious cycles leading to growth or decline of characteristics and behaviours. We use the term *feedback system* to refer to the set of elements and relationships involved in some feedback processes under consideration. A

✉ David C. Wynn
d.wynn@auckland.ac.nz

[1] University of Auckland, Auckland, New Zealand

[2] University of Strathclyde, Glasgow, UK

[3] Technical University of Denmark, Lyngby, Denmark

Published online: 28 March 2022

multitude of interacting feedback systems can be perceived in any design and development project.

More specifically, research literature indicates that feedback (viewed as goal-directed action and/or circular causal flows) influences design and development in many ways and at every process stage. For example, calculations, simulations and tests generate information based on which an emerging design is adjusted (Ahmed et al. 2003). Managers assign more resource when a project is seen to be falling behind schedule (Reichelt and Lyneis 1999). Data gathered by products in service are used to improve future product generations, and increasingly, to enhance products already in the field via software updates and data-driven features (Li et al. 2019). Feedback is essential in human-centered design because “without candid, actionable feedback from people, we won’t know how to push our ideas forward” (IDEO 2015). Feedback from users can also be instrumental in improving designs from one product generation to the next. More examples of feedback systems in the DDP can be readily identified (and many will be discussed throughout this article).

As well as being ubiquitous, feedback is usually seen as desirable in the DDP context—many design and development process models recommend frequent and rapid feedback as a strategy to reduce the scope of rework loops, to design in a complex and/or rapidly changing context, and to help ensure an emerging product will meet ambiguous stakeholder needs (see Wynn and Clarkson 2018, for a recent review of such models). More generally, researchers have shown that feedback can help to explain the emergence of design ideas (e.g. Glanville 2007b), the processes of design learning and collaboration (e.g. Sim and Duffy 2004; Dong 2004), and how development projects adapt in unexpected ways to new information and changes (e.g. Chiva-Gomez 2004). But at the same time, feedback processes in a project organisation cause resistance to desirable changes and dynamic complexity that makes comprehension and effective decision-making difficult (Senge 1990). And as DDP situations increase in scale, involving more participants, design issues and interdependencies, more interacting feedback loops can be found within them (Maurer 2017).

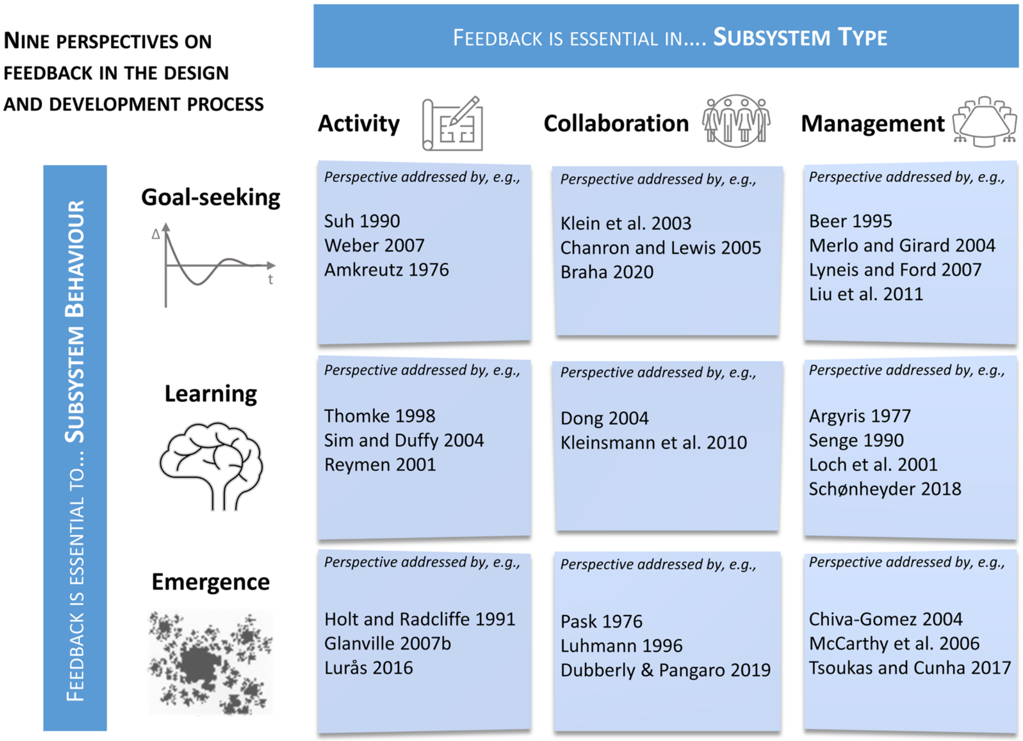
In summary, the importance of feedback in the DDP is widely appreciated in literature. But there are many perspectives with limited conceptual integration. In this article, we develop a conceptual framework and system-theoretic process model to synthesise some of the key ideas. The conceptual framework maps the insights that can be gained by appreciating the DDP in terms of the feedback systems within it. The process model integrates ideas from some previously distinct perspectives to indicate how the feedback essential to some important DDP system behaviours is realised. In particular and in contrast to many existing approaches, our framework and model incorporate selected insights from both (1) technocentric perspectives, in which DDP feedback is perceived by analogy to the feedback in an engineering control system or to natural feedback processes involving balancing and/or self-reinforcement; and (2) human-centric perspectives, in which emphasis is placed on the intelligent and reflective DDP participant in relation to the feedback system context. The integrated view presented in this article can help to more comprehensively frame DDP situations from a feedback systems perspective and to generate insight into those situations. We discuss several concrete examples that show how it can help to perceive opportunities for research and improvement.

2 Conceptual framework

The first contribution of this article, namely the conceptual framework, was developed by integrative literature review following a process similar to that used by Wynn and Eckert (2017). The scope for the review was set on publications that frame DDP situations in terms of feedback systems, emphasising feedback relating to human activity in the design and development process, and/or to design cognition itself. Work on feedback and related general systems principles that is not explicitly set in the DDP context was considered where appropriate to support the discussion, but is not comprehensively covered. Once the scope had been determined, literature was sought through a snowballing approach, beginning with internet search using relevant keywords, such as feedback, design, cybernetics, and so on. Key publication venues for engineering design and product development research were also searched and the search terms gradually expanded to include other relevant concepts. Because feedback is a well-established systems concept that has permeated much of design and development literature to some degree, there is an enormous number of directly or tangentially relevant publications.[1] But since the objective was to develop and elaborate a conceptual framework rather than to provide a systematic literature review, we made decisions on what to include and what to omit. Ultimately, the content of this article reflects our own perspectives on the topic.

Concurrent with literature study, a conceptual framework was proposed then iteratively developed. There were two main objectives. First, the framework was intended to

[1] Some researchers have argued that feedback is necessary in *any* system that performs a function. The TRIZ law of system completeness, for example, states that a system to perform a function must comprise four elements: a working unit to perform the task; an engine to drive it; a transmission between the two; and a control unit (the elements may involve people). This model has been applied to analyse systems in various domains including in business (Mann 2001) and biomimetics (Helfman Cohen et al. 2014).



Nine perspectives on feedback in the design and development process

Feedback is essential in.... Subsystem Type

Feedback is essential to... Subsystem Behaviour

	Activity	Collaboration	Management
Goal-seeking	*Perspective addressed by, e.g.,* Suh 1990 Weber 2007 Amkreutz 1976	*Perspective addressed by, e.g.,* Klein et al. 2003 Chanron and Lewis 2005 Braha 2020	*Perspective addressed by, e.g.,* Beer 1995 Merlo and Girard 2004 Lyneis and Ford 2007 Liu et al. 2011
Learning	*Perspective addressed by, e.g.,* Thomke 1998 Sim and Duffy 2004 Reymen 2001	*Perspective addressed by, e.g.,* Dong 2004 Kleinsmann et al. 2010	*Perspective addressed by, e.g.,* Argyris 1977 Senge 1990 Loch et al. 2001 Schønheyder 2018
Emergence	*Perspective addressed by, e.g.,* Holt and Radcliffe 1991 Glanville 2007b Lurås 2016	*Perspective addressed by, e.g.,* Pask 1976 Luhmann 1996 Dubberly & Pangaro 2019	*Perspective addressed by, e.g.,* Chiva-Gomez 2004 McCarthy et al. 2006 Tsoukas and Cunha 2017

Fig. 1 Nine perspectives on feedback in the design and development process, and selected publications that address them

express the range of DDP situations that have been treated from a feedback systems perspective. Second, the framework was intended to meaningfully organise a discussion of concepts from literature. It was decided to base it on a two-dimensional grid because this would provide structure for relating concepts while also being simple to depict. As literature study proceeded, the dimensions of the grid were refined concurrently with the review text and with reference to the objectives stated above. The framework was deemed complete once it was able to relate concepts from literature in a meaningful way, and once further reading ceased to prompt substantive changes. Finally, publications from the bibliography were revisited to check the accuracy of analysis.

2.1 Framework overview

The conceptual framework envisages a design and development process to take place in a complex dynamic system constituted by (1) the patterns of communication among the process participants, as well as (2) the patterns of connection that describe how each participant refers to and modifies certain design models, project documents and other representations as they work. On a more abstract level these patterns can be viewed as flows of information. The framework thus expands on the information processing view of project organisations (see, e.g. Galbraith 1974; Subrahmanian et al. 1997), in particular emphasising the dynamic and circular nature of many information flows involved. The framework also conveys the view that like other complex systems, design and development situations can be appreciated from different standpoints and therefore, the roles of feedback can be perceived in different ways.

The framework relates nine perspectives or ‘windows’ onto feedback in the DDP by organising them along two dimensions as shown in Fig. 1. The first dimension draws on the observation that a variety of subsystems can be perceived in most design and development situations, allowing for different views on the role of feedback. We consider three types of subsystem, reflecting some of the main topics treated in the reviewed literature:

- *Activity subsystems* and specifically the processes by which individuals (or groups when viewed as acting in unison) complete design and development tasks.
- *Collaboration subsystems* and specifically the processes by which team members and other stakeholders work together to complete design and development tasks.
- *Management subsystems* and specifically the processes of managing design and development work, as done by individuals managing their own work through to the management processes in large-scale projects.

The second dimension of the framework concerns some dynamic behaviours of the DDP, in which feedback has important roles:

- *Goal-seeking* concerns how a subsystem can approach its goals and can absorb uncertainties and disturbances. The value of goal-seeking to absorb uncertainty is exemplified by a thermostat, which brings a room to a desired temperature (the goal) although the outside temperature and other system parameters are subject to variation. Uncertainty is endemic in the design and development process—examples of goal-seeking and uncertainty absorption in this context include adjusting a design solution to approach specifications in response to unforeseen test or analysis results, and reallocating resources to ensure timely project delivery under unpredictable workload fluctuations.
- *Learning* concerns how a subsystem can develop knowledge and adjust its behaviour, including in response to a changing or ill-structured context. This is essential in design and development at every level, due in part to the inherent uniqueness of the work needed to realise a new design.
- *Emergence* concerns how a subsystem can generate unexpected trajectories—for instance, how new designs can be created and how work can be reconfigured to take advantage of new insights and opportunities. Emergence can also lead to adverse situations such as cost overruns and design flaws.

All these subsystem types and behaviours can be perceived in most design and development situations, and at different scales. Hence, the nine perspectives of our framework provide windows onto DDP feedback that allow it to be viewed from different, but interrelated standpoints.

While other ways to structure an analysis would be possible, this framework proved useful to relate key concepts from literature and also to demonstrate how feedback is intertwined through many aspects of the design and development process. The nine perspectives are further clarified by discussion of relevant work in the next subsections. The work to be discussed ranges from highly conceptual and discursive in nature, through to research based on mathematical and computer modelling of DDP feedback situations. Finally, it should be emphasised that the framework was developed to relate concepts, not to classify literature, therefore it is not uncommon for a publication to address multiple perspectives.

2.2 Feedback in DDP activity

Considering the first column of Fig. 1, design and development activity takes place under incomplete or imperfect information and involves adjustments, learning and emergence as new information becomes available. The involvement of feedback in such activity is recognised in almost all design process models. For instance, most stage-based models of engineering design indicate loops from later project stages to earlier ones, typically described as feedback (e.g. French 1999; Pahl et al. 2007) and/or iteration (e.g. Hubka and Eder 1988). Also suggestive of feedback, several theories of innovative designing involve cycles of influence connecting different spaces or domains under consideration by the designer. For instance in C–K theory, knowledge (K-space) is needed to propose and modify concepts (C-space), while consideration or testing of concepts prompts the acquisition of new relevant knowledge (Hatchuel et al. 2004; Hatchuel and Weil 2009). In another example, Parameter Analysis presents innovative designing as a series of transitions between concepts and their realisation in physical configurations, that influence one another (Kroll 2013). Other models describe cycles between the forms of logic involved in analysis and synthesis (e.g. March 1984; Kroll and Koskela 2016).

These are just a few examples—cyclic flows of influence are central to many publications on design theory and design processes. But such work does not usually emphasise systemic issues that arise from the depicted feedback loops, such as the influence of information quality and timeliness on convergence and stability, the existence of vicious and virtuous cycles, and many other feedback-related concepts to be considered in this article. Many design and development process models also do not clearly distinguish (1) feedback, in the sense of cybernetic (self)regulation or circular flows of influence as analysed in this article, from (2) iteration, described by Wynn and Eckert (2017) as the revisiting of tasks, issues, or aspects of a design for a variety of reasons.

The next subsections work down the first column of Fig. 1, considering how feedback allows DDP activity to proceed in a progressive and iterative way that involves goal-seeking, learning and emergence of unforeseen outcomes. To reiterate, we do not provide a comprehensive literature review, but a discussion of key concepts that shows how feedback is essential to DDP activity in a variety of ways. The discussion focuses on work in which feedback and its systemic implications are central topics.

2.2.1 Feedback and goal-seeking in DDP activity

In one seminal publication that discusses feedback and goal-seeking in design activity, thereby addressing the top-leftmost cell of Fig. 1, Simon (1981) describes designing in terms of generate-test cycles in which alternatives are proposed then evaluated against relevant goals and constraints. He relates this to handling the incompletely decomposable nature of design problems, stating that generating

$$\frac{Y}{X} = \frac{G}{1+GH} \approx \frac{G}{GH} = H^{-1}\ for\ GH \gg 1$$

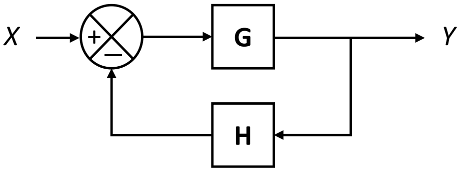


Fig. 2 The design process as a feedback control loop. Figure redrawn from Suh (2005). License: Creative Commons BY-NC-SA

alternatives "implicitly define[s] the decomposition of the design problem", while the tests provide feedback ensuring that dependencies are properly accounted for. He also suggests these generate-test cycles are nested, reflecting the nested levels of a design problem being addressed. Later work expanded on this observation that feedback is central to the goal-directedness of design activity. Suh (1990, 2005), for example, depicts the engineering design process as a feedback loop between a creative process and an analytical process, shown as *G* and *H* respectively in Fig. 2. Suh emphasises the importance of analysis to supplement creativity, arguing that "the gain of the feedback loop should be as large as possible to converge to a correct solution quickly; that is, the ability to judge the outcome of the creative process improves the creative process itself." Suh presents the model briefly and without elaboration, for example, the signals labelled as *X* and *Y* in Fig. 2 could be more clearly defined. Also, some important aspects of the design convergence process are not explicitly represented in this model, including the increasing amount of information that must be considered as design moves forward and the progressively reducing room for manoeuvre (see, e.g. Reich 2008; Ullman 2015, for discussion on these topics). In a similar model, Weber (2007) suggests that the engineering design process can be modelled in a "strongly abstracted representation" as a dynamically driven control circuit. He writes that design analysis methods are analogous to sensors, helping designers to detect the gap between desired and actual properties of the design, while design synthesis methods are analogous to actuators, helping designers to adjust the design in the desired direction. In Weber's model, external conditions impacting the design process, such as changes in drivers and constraints, are viewed as disturbances to be absorbed.

Feedback and its role in goal-seeking systems more generally have been studied extensively in the field of cybernetics. In overview, cybernetics is concerned with understanding how systems are, or can be controlled or influenced to account for the presence of uncertainty, disturbance and changing objectives (Wiener 1948). On the one hand, these ideas have been applied quantitatively, for instance to the study of regulation and control in physical systems (noting that servomechanisms, such as mechanical governors and thermostats, had been developed earlier). Other researchers have developed cybernetic ideas conceptually and discursively, for instance to analyse societal and organisational situations (e.g. Geyer 1995; Umpleby 1997; Beer 1995) and the processes of cognition and learning (e.g. Pask 1969; von Foerster 2003), as well as many other types of system. The conceptual application of cybernetic principles is often appropriate where the subject matter is complex and rich in ambiguity.

Although the work mentioned at the start of this subsection suggests that engineering design activity can be viewed as goal-seeking driven by feedback, the mentioned publications do not explicitly draw on insights into feedback from cybernetics research. Some other engineering design researchers have connected the topic with classical cybernetics but without emphasising feedback. For example, according to the German engineering design guideline VDI2221, "a breakdown of the problem-solving process into parallel paths is recognised by cybernetics as being an effective and economical strategy for solving complex problems" (VDI2221 1987). However, this argument is not further elaborated in the guideline. Hubka and Eder (1996) acknowledge the impact of cybernetics on the development of design science, in particular by inspiring the distinction between functional and structural perspectives on a system. A comprehensive system description should incorporate both perspectives (see e.g. Pask 1969, for discussion). But Hubka and Eder (1996) do not discuss cybernetics in depth and key aspects, such as goal-directedness and the focus on dynamic behaviours, are not emphasised. A connection between cybernetics and the well-known theory of technical systems is also briefly indicated in their later work (Hubka and Eder 1988), but again, a detailed analysis is not presented. These publications perhaps refer to a systems perspective in general rather than to the insights into dynamic systems and feedback that are offered by classical cybernetics. Eder (1998) is more specific, writing that feedback principles help to explain why successful control of the design process can be achieved without a detailed understanding or model of its working mechanisms. The existence of feedback is why, he proposes, systematic methods can be devised to guide design, even while design cognition itself remains incompletely understood.

Few detailed accounts of engineering design activity explicitly integrate insights into feedback and goal-seeking from classical cybernetics research. In one such analysis, Amkreutz (1976) represents designing as a generation function coupled to a feedback function, similar to Suh's model discussed earlier (Fig. 3). The generation function *G* depicted at the bottom of Fig. 3 is responsible for generating the design solution, considering all relevant input information i_G—most importantly the constraints, used to determine

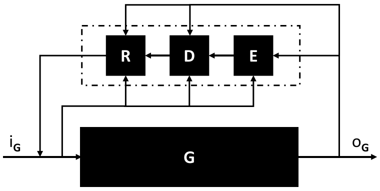


Fig. 3 Design process as a control system. *G* represents the design generation function. The dashed outline represents the feedback function, that is decomposed into evaluation *E*, decision *D*, and regulation (i.e. design modification) *R*. Reproduced from Amkreutz (1976) with permission from Elsevier

whether a design is feasible, and the criteria, used to rank feasible designs in relation to one another. The feedback function, depicted as a dashed outline in Fig. 3, evaluates the design and drives the evolution of constraints and criteria during the design process, in tandem with refinement of the design solution. Amkreutz (1976) further decomposes the feedback function into evaluation *E*, decision *D*, and regulation *R* (that involves changing the design to enact the decision). He writes that designing involves numerous coupled control systems, each treating a different design issue or discipline. Lhote et al. (1999) further expand on the coupled control system analogy by conceptualising the design process as a series of loops in which each actor produces a plan for the next and controls that plan (and adherence to it) by feedback.

Overall, work discussed in this subsection establishes that design activity (in particular engineering design) can be perceived and modelled as a goal-seeking process in which technical characteristics are adjusted to iteratively converge on design goals. But the models mainly focus on the process of adjusting design characteristics, and do not emphasise how feedback influences the designer's knowledge and decisions. Also, their presentation of design as a convergence process does not fully account for how novel and creative solutions are reached. The next subsections discuss feedback-based models that address these latter aspects of design activity.

2.2.2 Feedback and learning in DDP activity

This section concerns the second behaviour in the leftmost column of Fig. 1, namely learning in DDP activity. Learning is essential to overcome imperfect or incomplete knowledge while designing (Sim and Duffy 2003), and, more generally, is also central to the intelligent behaviours of process participants. It can involve the generation of new knowledge or the modification of existing knowledge, and occurs concurrently with and intertwined with design activity (Andreasen et al. 2015). Because learning causes knowledge to evolve according to the designer's unique situation, its study can help to appreciate why designers respond in different ways to the same information, and why the same designer might respond differently to similar information received at different times. The rest of this section demonstrates the close relationship between learning and feedback in DDP activity.

Feedback in the learning process has been extensively discussed in design and development literature. To provide a few examples: Pahl et al. (2007) write that learning from feedback increases the amount of knowledge available during design, and "information feedback must be repeated until the information content has reached the level at which the optimum solution can be found." Thomke (1998) builds on the views of Simon (1981) and Clark and Fujimoto (1989) when writing that engineering design comprises interlinked design-build-test cycles, in particular discussing the cycle of "trial, failure, learning, correction and retrial" (suggesting a corrective and learning-driven feedback loop) used to improve design performance or to detect and eliminate flaws. Sim and Duffy (2003) connect design learning to goal-seeking, writing that learning occurs during the elaboration and reformulation of ill-specified goals, and leads to improved knowledge of how to approach those goals. In their later work, Sim and Duffy (2004) state that design learning can be viewed as a goal-directed feedback system in itself—they view the goals for feedback-driven learning to include design improvement, dealing with novelty, and avoiding design failure. While the above discussion mainly relates to design tasks, learning and its relation to feedback has also been considered in DDP activity more broadly. In one such contribution, Wynn et al. (2010) develop a cybernetic model to show that the purpose of design process modelling is to generate knowledge that will inform feedback decisions. For example, after observing that a design process requires improvement, an activity network model might be developed to generate knowledge about the underlying reasons, prompting changes to bring the process closer to its desired state.

Most of the work discussed in the previous paragraph idealises (explicitly or implicitly) DDP participants as rational agents as defined by Newell (1981)—i.e. as agents that evaluate each decision by considering which of the possible options is most appropriate to achieve their goals, based on their knowledge. In contrast, other authors focus on the role of reflection in design learning. Reymen (2001) for example defines reflection as introspective contemplation on the design situation and recent actions, considering issues such as whether the right problems are being addressed and in the right way. The insights gained influence future design actions, forming a feedback cycle. Reymen (2001) argues that reflection during design can lead to a steeper learning curve and improved design output. Schön (1983) considers that reflection occurs when a designer sees implications of their emerging design that were not intended and consciously judges them to be desirable or undesirable, leading to adjustments in how they frame the design situation for the next iteration. This can also be viewed as a form of learning in which the designer evolves their way of looking at the design situation. Valkenburg and Dorst (1998) show that designers' reflection strategies appear important to design process direction and design performance, while Reich and Subrahmanian (2020) develop the PSI framework to guide reflection on a wide range of design situations (that could be at any level of our framework) by considering issues in three spaces: the problem itself (P), the people and perspectives involved in addressing the problem (S) and the institutional, organisational and process issues involved in addressing the problem (I). The importance of feedback and thoughtful reflection in the learning process have also been researched extensively in the context of education. For example, intentional self-regulation of the learning process is thought to lead to improved learning performance. Feedback information for learning about the subject, and for learning how to self-regulate the learning process itself, can be provided externally or can occur in relation to a student's individually-formulated criteria, such as performance against plans (see, e.g. Butler and Winne 1995, for a review).

To summarise, learning is closely intertwined with design activity and occurs through feedback processes. Some researchers suggest that these processes are essentially rational, while others emphasise the importance of reflective judgement. In either case, learning impacts the knowledge that informs a process participant's response to feedback information. Learning is necessary to incorporate in a feedback-centric model of the DDP if it is to explain how the response to a design situation will differ depending on when and by whom it is encountered and also, if it is to convey design and development as a knowledge-generating and thoughtful process.

2.2.3 Feedback and emergence in DDP activity

Moving on to the bottom-leftmost perspective of Fig. 1, the aim of this subsection is to establish that feedback is closely involved in generating emergent DDP outcomes. We adopt the view that an emergent outcome is one that cannot be predicted by an observer or participant in a system. This encompasses both weak emergence (outcomes that could, in principle, be deduced from knowledge of the system parts and interactions) and strong emergence (outcomes that could not be deduced, even in principle). In a design and development context, emergence can arise in a system that is being designed or intervened with. Emergence can also be observed in the design process itself. For example, different designers may take different routes through the design process and may arrive at different solutions according to their individual knowledge and experience (Braha and Maimon 1998). Emergent situations can cause subsequent DDP activity to react in unpredicted (also emergent) ways. The outcomes of emergence in DDP activity can be desirable (e.g. novel solutions) or undesirable (e.g. schedule overruns and design flaws).

Feedback is involved in DDP emergence in several ways. First, it can be noted that although goal-seeking is associated with maintaining stability and approaching goals, this is not at odds with one of the most important properties of the design process—its ability to produce creative and unexpected solutions. Many models of designing that focus on the creation of novel solutions present the design process as an interplay between divergent and convergent activities (e.g. Council 2007; IDEO 2015). While the models of feedback in DDP activity discussed in Sect. 2.2.1 focus mainly on convergence, feedback also plays a role in divergent activities, which are equally important in creativity. In particular, a variety of methods and cognitive strategies have been developed to prompt divergent thinking towards creative ideas (see, e.g. Finke et al. 1992). Feedback in divergent design activities could accordingly be perceived in the self-regulation of those activities with reference to desirable practices.

Second, learning, which is closely connected to feedback as discussed in the previous subsection, can also lead to emergence. This is because a designer's knowledge influences how they interrogate their emerging design, how they interpret the results, and how they decide to adjust the design on the next iteration. Since the knowledge being evolved through learning is idiosyncratic to the designer, and since a design situation yields different information depending on how it is perceived and interrogated, the process produces unexpected outputs that can be viewed as emergent. Taking a different perspective, Schön (1983) views the essence of designing as dealing with unique situations, which as mentioned in the previous subsection is achieved by deciding which aspects of the design situation are to be focused on (framing), making changes (moving), and becoming aware of unanticipated consequences of those changes (seeing)—which may trigger a deliberate reframing of the situation for the next design step. Schön (1983) describes this as a reflective conversation between designer and situation. Because previous decisions influence future ones as the designer consciously constructs the situation in which they work, this perspective also suggests connections between circular flows of influence and emergence in design activity. To summarise, while both views discussed in this paragraph show the role of feedback in design emergence, they differ in that the first presents emergence as arising from essentially-rational situated activity, while the latter emphasises the ill-defined nature of design goals and the importance of the designer's thoughtful practice in deliberately constructing their perceptions of a design situation (see also Dorst and Dijkhuis 1995).

Third, cybernetics theory has been used to connect feedback in design activity to design emergence. For instance, Suh (1990) proposes that the ideas of classical cybernetics could help to explain how the generation and evaluation functions of designing lead to novel designs when connected together in a feedback cycle. He draws an analogy to how chaotic nonlinear dynamics can emerge from individually well-understood functions when connected in this way. Another cybernetic perspective on design activity that emphasises its emergent characteristics is offered by Glanville (2007b), who explains how some advanced concepts of second-order cybernetic theory (that is to be further discussed in Sect. 2.3.3) could together provide the basis for an account of designing that emphasises its inherent creativity and novelty. However, even in context of this subsection, the work is highly conceptual and as Glanville himself suggests, could be considered "romantic". Finally in this subsection, researchers including Holt and Radcliffe (1991) and Lurås (2016) have considered how Vickers' appreciative systems theory (closely relating to second-order cybernetics) can offer insight into emergent situations in design. In overview, this theory views an individual's understanding of a situation as being built by framing the "flux of events and experience" using their *appreciative setting* (Vickers 2010). The appreciative setting is an accumulation of experience and includes ways of seeing what is relevant as well as norms that concern what is desirable, and that guide actions. Those actions in turn affect the flux of events and experience creating a reciprocal flow of influence, i.e. a feedback loop. Holt and Radcliffe (1991) draw on appreciative systems theory to present a theoretical account of how learning leads to design emergence. They argue that new information that arises while designing can either support the appreciative setting or can disrupt it, causing instability and perhaps a new appreciative system to emerge. They associate the latter with a desirable "dramatic change in the richness and variety of ideas."

To summarise this subsection, a number of researchers have shown that feedback can help to explain how novel designs emerge from DDP activity, and how that activity responds to emerging designs in unexpected ways. One of the main consequences for this article is that a feedback systems perspective is not limited to convergent situations, as are common in routine or parametric engineering design, but also casts light on creative and nonroutine design activity in less constrained contexts. In critique, the work discussed in this subsection is highly conceptual and mostly does not provide concrete recommendations for practice.

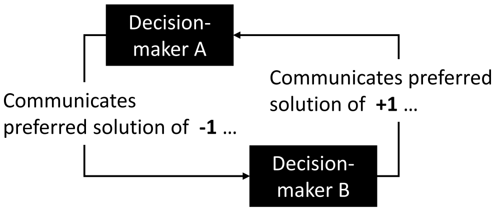

Decision-maker A
Communicates preferred solution of **+1** ...
Communicates preferred solution of **-1** ...
Decision-maker B

Fig. 4 The simplest possible cyclic asymmetric network representing two coupled decision makers who demand mutually unacceptable solutions, leading to oscillation. Illustrates concepts discussed by Klein et al. (2003)

2.3 Feedback in DDP collaboration

We now move on to discuss the second column of the framework depicted in Fig. 1, which concerns the roles of feedback among collaborating DDP team members.

2.3.1 Feedback and goal-seeking in DDP collaboration

Beginning with the topmost cell of the second column in Fig. 1, one of the challenges in approaching goals (i.e. converging on a solution) in team design work is that it is not possible to perfectly decompose a typical design problem into subproblems to be addressed by each team member. Therefore, the objectives and constraints faced by team members are interdependent. In practice, integrating interdependent design contributions is typically achieved through an iterative process of piecewise adjustments, ideally of decreasing magnitude (Wynn and Eckert 2017). This process has been examined in design negotiation research, which considers how self-interested design agents should behave locally to reach a globally satisfactory outcome with a minimum of iterations. Klein et al. (2003) illustrate some of the issues by discussing a simplified situation in which two goal-seeking agents with conflicting objectives are coupled as depicted in Fig. 4. In this example, Agent A receives a proposed solution from agent B (labelled +1) and finds it unsatisfactory, so provides feedback that the solution should be changed to – 1 to meet their own objectives. Agent B then provides the opposite feedback. In this very simple case, and many others involving intercoupled cycles and delays, the system oscillates without converging to a mutually satisfactory solution in which all the goals are adequately met. Klein et al. (2003) suggest several negotiation strategies that can assist convergence in such situations, including mediation, supplementing the feedback information with indications of commitment to a particular solution, and decomposing the design process into stages or modules to decouple some of the feedback loops, among other strategies. Some researchers apply game-theoretic approaches to study design negotiation (e.g. Lewis and Mistree 1997; Hernandez et al. 2002; Chanron and Lewis 2005). Chanron and Lewis (2005), for example, investigate the properties of design problems that allow design convergence under decentralised decision-making. They show that the problem structure and the objective functions sought by each designer determine whether a Nash equilibrium exists, and hence whether the back-and-forth feedback process can converge.

As the number of interdependent nodes in a decentralised decision system increases, feedback loops can increase in scale to encompass multiple decision-making nodes, and individual nodes can participate in multiple loops. Braha and Bar-Yam (2007) develop a computational model that is similar in abstraction level to Fig. 4 (but with many more nodes) to show that certain statistical properties of real-world product development task networks, such as a small-world structure, tend to accelerate design convergence. The few nodes that are highly connected in such networks also have greater influence on problem-solving dynamics than the many that are not. Expanding on these insights, Braha (2020) shows that coordination in real design problem-solving networks is aided by structures which allow small clusters of problems to be resolved rapidly before being integrated into the solution of larger subproblems, that are governed by loops with slower dynamics. This work offers insights into the dynamics of intercoupled feedback loops that involve large numbers of problem-solving nodes, however, does not discuss the behaviour of the individual feedback loop or decision-making node in depth.

Overall, the work discussed in this subsection establishes the role of feedback in distributed design convergence, in particular for heterarchical situations (i.e. with weak or no overarching control) in which design is subdivided among multiple decision-makers responsible for different, but interdependent, objectives and parameters. Such situations are common in practice, especially in large-scale design and development.

2.3.2 Feedback and learning in DDP collaboration

Considering the central cell of Fig. 1, feedback and learning in DDP collaboration processes are interrelated in several ways. For instance, Klein et al. (2003) write that the negotiation of solutions within the feedback processes discussed in the previous subsection is aided when agents learn by adapting their decision-making to recognise the strategies of other agents and previous solutions to similar problems. Other work concerns the development of knowledge in a design team context, considering that team participants have unique perspectives (Bucciarelli 2002; Kleinsmann et al. 2010) and that a dynamic learning process takes place to bring together the different perspectives such that their work is brought into alignment. Such processes have been described as social in nature and occur throughout a project, although may be more important at the outset because this is when a shared understanding first needs to be developed (Kleinsmann and Valkenburg 2008). The task is likely to be more difficult when many specialists from different disciplines are involved.

Some authors draw on cybernetics theory to cast light on how feedback relates to learning in DDP collaboration. Dong (2004), for example, draws on second-order cybernetics in his theoretical account of design collaboration, emphasising that design processes should be understood on two levels: the cognitive processes of designers engaged in design activity; and the social processes that explain how designers develop congruent thinking, that he suggests can be viewed in second-order cybernetic terms. In other words, he considers that the goals of a collaborative design process, that are approached through feedback, include both design problem solving and the development of shared understanding. In later work, Dong (2005) supports this perspective by showing empirically that the process of bringing perspectives into alignment is reflected in increasing congruency of language used during a design team discussion as it progresses. Maier et al. (2012, 2014) discuss how design team work can be seen as a cybernetic system with an emphasis on modelling activity, which they see as a form of learning and as central to designing. They perceive a design team to be embedded with an ecosystem of models and the design problem in a designing system, which is also a modelling system governed by feedback. They describe how the ecosystem of models within such a system, which includes product and process models as well as design methods and mental models, is interpreted by the design team to regulate design processes towards goals. They also consider that these models are evolved over time through learning, and that different perspectives on those models exist within the team—in fact these are necessary to handle the complexity of designing. While emphasising that individual team members construe models differently and must negotiate during the design and learning processes, Maier et al. (2012) do not unpack the interactions among individual team members. Discussing this and related work, Andreasen et al. (2015) write that a cybernetic perspective can provide insight into conceptual design in teams but also observe that "cybernetic models of design do not really show what is synthesised and manipulated when designing."

2.3.3 Feedback and emergence in DDP collaboration

This section discusses the bottommost cell of the second column of Fig. 1, in particular how feedback allows a team working together to generate design outcomes that they do not expect, and that extend beyond what they would produce as individuals. One body of theory that offers relevant insights is second-order cybernetics, which arose from the reflexive application of cybernetic theory to explain itself and more generally, involves the application of cybernetics to appreciate the processes of observing systems and the observers themselves (von Foerster 2003) (see also Scott 2004, for a historical introduction). This is a conceptually rich research area that impinges on several perspectives of our framework; a comprehensive discussion of its implications for the design and development process is outside the scope of this review. Some of the perspectives are collected in Fischer and Herr (2019). The following paragraphs outline selected concepts that are relevant to our discussion of emergence in DDP collaboration.

One implication of adopting a second-order cybernetic perspective is that an observer should be recognised to be dynamically coupled to the situation being observed through reciprocal flows of influence, such that the two constitute (in the terminology of this article) a feedback system. For instance, when a design process participant considers the work of a colleague, their observations affect their own design decisions and through interdependencies among decisions, the colleague's work is affected in turn. Such dynamic couplings contribute to unpredictable, i.e. emergent outcomes. Pask's conversation theory (Pask 1976) is one important contribution to second-order cybernetics that can offer further insight into emergence in design collaboration, through a deeper analysis of the couplings described above. More specifically, conversation theory focuses on how understanding is constructed through cycles of feedback interactions between participants in a conversation, such as occurs in design teams. It emphasises that conversations do not involve direct transmission and reception of information, but are formed from back-and-forth utterances of language—and because language is ambiguous there is no precise coding scheme to extract meaning from it. The listener must therefore frame each utterance according to what they thought the speaker was trying to say, to form meaning. As the conversation proceeds, each participant gains insight into the others' understanding, leading to adjustments such that a more congruent understanding is socially constructed (Scott 2001). Pask (1976) provides a detailed formalism for modelling this process, while research in design cybernetics considers that design activity in the most general sense can be conceptualised as a kind of conversation between a designer and the design situation they are embedded with (Glanville 2007b; Dubberly and Pangaro 2019).

One way that second-order cybernetics and conversation theory offer insight into how feedback leads to emergence in DDP collaboration is in the emphasis that the interpretation of each communication to form meaning is determined by each individual's mental model, which is evolved through feedback processes and unique to them. Addressing this idea in a general context, Luhmann (1995) writes that meaning, which he defines as the horizon of possibilities information implies to a particular individual, is unique to each participant in the communication process. An observer cannot know precisely what the communication means to a particular individual, and consequently cannot predict exactly in advance how they may react to it—although they can observe what response it elicits (Luhmann 1995). A person's response to successive similar events may furthermore differ as they learn and evolve the models that govern how they interpret their observations, as well as how they respond to them. To summarise the implications for this subsection, communications and meaning during the collaboration process are not absolute because they depend on idiosyncratic mental models, while those models are themselves contingent on prior communications. According to this perspective, mental models and meaning are dynamically coupled and constructed through feedback loops among team members. The feedback-driven coevolution of mental models may be conceptualised as the mechanism by which DDP participants construct a reality on several levels: they each construct a personal understanding of their context; together they construct the processes by which they interact, and through those processes they construct the design itself, as well.

Overall, because systems of the type described above are have internal memory and are intercoupled, they are what von Foerster (2003) describes as "non-trivial machines," implying systems whose trajectories cannot be predicted. Such a system-theoretic perspective resonates with discussions in other areas of design research, for example, ambiguity (scope for interpretation) in design representations and communication has been viewed as essential to promote creativity in teams because it is important to maintain a degree of unfixedness in design communication, to prompt each designer to reinterpret communicated ideas, and to allow them to project their own thoughts onto those unfixed ideas (Minneman 1991).

2.4 Feedback in DDP management

Moving on to the rightmost and final column of the framework in Fig. 1, design and development process management occurs in DDP situations at all scales. For instance, the roles of feedback in self-regulation of learning and in design negotiation situations have already been mentioned in Sects. 2.2.2 and 2.3.1 respectively. Other research relevant to this topic has focused on management in large-scale concurrent engineering projects in which work is partitioned among several teams. Such projects involve many interdependent issues, tasks and individuals (Eppinger et al. 1994) and hence, complex networks of information flow within which feedback can be perceived. The following subsections focus mainly, but not exclusively on feedback in the large-scale DDP management context.

2.4.1 Feedback and goal-seeking in DDP management

Uncertainty is a significant issue in the design and development process (Pich et al. 2002; Yassine 2007; Wynn et al. 2011). In consequence, unplanned events are inevitable. With regards to the top-rightmost cell of Fig. 1, a number of authors have shown that feedback mechanisms within a project are necessary to coordinate its response to such events and hence, to absorb the uncertainty while still meeting project goals. For instance, well-known project management approaches, such as Agile and the spiral model, are structured to ensure regular feedback, so that issues can be identified and responded to in a timely way (e.g. Unger and Eppinger 2011). While feedback is therefore recognised as desirable to achieving DDP goals in the presence of uncertainty, intertwined feedback loops are also significant drivers of dynamic complexity in large-scale projects (Ramasesh and Browning 2014).

Recognising the importance of feedback when managing a DDP, numerous models have been developed to explore goal-seeking mechanisms in this context. The following paragraphs discuss three main subcategories: qualitative models, modelling methodologies, and quantitative approaches.

Qualitative models The first subcategory concerns descriptive and prescriptive models emphasising the role of feedback in the DDP. For instance, Wellsandt et al. (2018) elaborate a number of situations where feedback is a prominent driver of design iteration, including the gathering of information in support of quality management, gathering of customer feedback on an emerging design, and the transfer of insights from in-service to a new design. They also distinguish "active" feedback in which information is sought by the developers from "passive" feedback, in which information is provided from outside the development project and must be responded to. Wellsandt et al. (2018) point out that feedback increases the amount of information in a project, but the information may not be ideal to decision needs, may be conflicting, and may be overwhelming if project participants are not able to filter it effectively. In another qualitative model, Huang and Gu (2006) write that feedback is necessary to absorb uncertainty that arises from cognitive limitations and the innate complexity of design activity. Without feedback, they argue, a DDP will become unstable with undesirable impacts on cost, quality and time. They propose the application of feedback control theory to analyse and optimise a development process, but do not explain in detail how this could be achieved. Vajna and Burchardt (1998) propose that a product development process can be perceived as a structure of "autonomous and self-checking" cells that are each responsible for certain process activities, arguing that feedback and "cross-controlling" among the cells enables "automatic regulating and permanent optimisation." But again, they do not elaborate these ideas in depth.

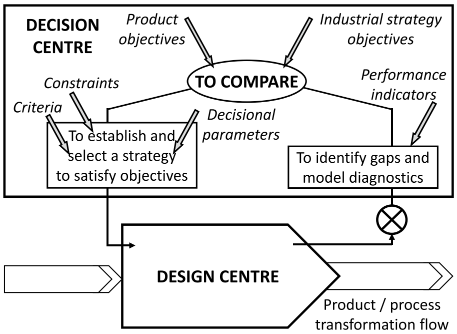
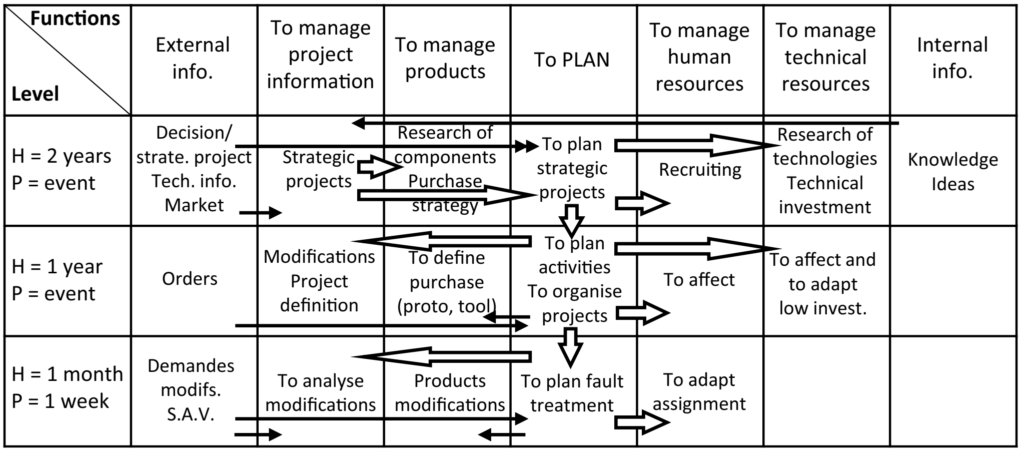

DECISION CENTRE
Product objectives
Industrial strategy objectives
Constraints
Criteria
TO COMPARE
Performance indicators
Decisional parameters
To establish and select a strategy to satisfy objectives
To identify gaps and model diagnostics
DESIGN CENTRE
Product / process transformation flow

Fig. 5 GRAI reference model indicating the decision system (top) responsible for control of the technological system (bottom) in product development. Reproduced from Merlo and Girard (2004) with permission from Elsevier

Some of the most established qualitative models that emphasise the importance of feedback in large-scale design and development processes focus on its role in coordination. As already mentioned, responsive coordination is important in the DDP to identify and respond to unplanned events, for instance by redefining, reprioritising and rescheduling tasks, reallocating resources, and conveying updated information to the appropriate parties. The greater the complexity and novelty in a project, the more difficult it is to plan, the more unplanned events occur, and the more management attention and responsive coordination is typically required (Galbraith 1974; Christiansen 1993). One qualitative model that emphasises the role of feedback and goal-seeking in coordinating the DDP is the GRAI reference model of Doumeingts et al. (1996). As depicted in Fig. 5, this model conceptually divides the DDP into three subsystems: a technological system which emphasises design workflow; a decision system that emphasises management; and an information system that supports and connects the two. The technological system comprises autonomous design centres that periodically provide updates so that performance gaps can be detected by the decision system and necessary changes made (Merlo and Girard 2004). Girard and Doumeingts (2004b) write that a well-structured decision system is important to controlling design performance despite inherent uncertainties and complexities.

In another qualitative model focusing on the role of feedback in DDP coordination, Wilberg et al. (2015) interpret the Viable Systems Model of Beer (1995) in the context of engineering change management. In overview, the Viable Systems Model is based on the argument that subsystems within a firm each have (or should have) the property of viability (Beer 1995). A viable system uses internal feedback processes to autonomously respond to change, manage its



Functions / Level	External info.	To manage project information	To manage products	To PLAN	To manage human resources	To manage technical resources	Internal info.
H = 2 years P = event	Decision/ strate. project Tech. info. Market	Strategic projects	Research of components Purchase strategy	To plan strategic projects	Recruiting	Research of technologies Technical investment	Knowledge Ideas
H = 1 year P = event	Orders	Modifications Project definition	To define purchase (proto, tool)	To plan activities To organise projects	To affect	To affect and to adapt low invest.	
H = 1 month P = 1 week	Demandes modifs. S.A.V.	To analyse modifications	Products modifications	To plan fault treatment	To adapt assignment		

Fig. 6 GRAI grid showing decision system interactions in a fuel equipment manufacturer. H refers to the horizon of the decision, i.e. time over which it remains valid, and P represents the period, i.e. time before the decision is reconsidered. Reproduced from Girard et al. (1999) with permission from Springer Nature

resources, and manage interactions with its environment, thereby maintaining its existence in a context of uncertainty without requiring external control, within limits. The Viable Systems Model posits that to be viable, any system must comprise five necessary subsystems, each of which has a specific regulatory role. Wilberg et al. (2015) apply this model to analyse the engineering change management system in a construction project. The real-world system was mapped against the five subsystems of the Viable Systems Model, and gaps were identified to generate recommendations for improvement. Elezi (2015) applied the model to analyse and design management control systems in the DDP. He argues that such systems are typically described as homeostatic regulators by analogy to engineering servomechanisms, but in fact should be treated as viable systems. Overall, the Viable Systems Model offers valuable insight into the feedback structures in organisations, but has sometimes been described as esoteric and difficult to grasp for those unfamiliar with cybernetic theory (Brocklesby et al. 1995).

Finally in this subcategory, O'Donnell and Duffy (2002) developed a model to emphasise the role of feedback and goal-seeking in managing performance of engineering design processes. In their model, design and management activities are presented as being closely intertwined at every level of the DDP. Effectiveness of a design activity is defined as how well design goals are achieved, and effectiveness of the management activity is defined as how well process goals, primarily time and cost, are achieved. The model suggests both of these performance measures drive a feedback control loop in which goals are adjusted or resources are reallocated. O'Donnell and Duffy (2002) also indicate that tradeoffs must be made among achieving design and management goals, and emphasise the importance of using performance indicators that are appropriately aligned to goals.

Modelling methodologies The second subcategory comprises modelling approaches to help practitioners model and understand feedback processes relating to goal-seeking in their own situations. For instance, the IDEF0 function modelling approach decomposes a DDP into functions, each of which is characterised by inputs, outputs, controls and mechanisms (USAF 1981; Colquhoun et al. 1993; Romero et al. 2008). This explicitly requires the modeller to consider the importance of how key functions are controlled, which in many situations will involve feedback alongside other mechanisms, such as planning and goal-setting. However, the individual feedback loop is not explicitly emphasised in IDEF0. Another modelling approach emphasising feedback is the GRAI methodology, which builds on the GRAI reference model discussed above to provide a process for modelling a DDP organisation using several coupled representations to understand and improve it. Of these, the most relevant to this article is the GRAI grid. This is a diagramming approach that prompts the modeller to consider the interactions among different control mechanisms within the product development decision system, each of which have different characteristic timeframes and scopes of action (Girard and Doumeingts 2004a). To illustrate, an example GRAI grid is shown in Fig. 6. The GRAI approach considers that modelling the decision system in this way can help to structure and coordinate the decision processes (Girard and Doumeingts 2004a), which is thought to be a precursor to designing an information system that ensures the correct information is available to effectively coordinate among the decision-making levels (Merlo and Girard 2004).

Many of the models discussed so far view a process on a level of abstraction at which feedback happens, in essence,

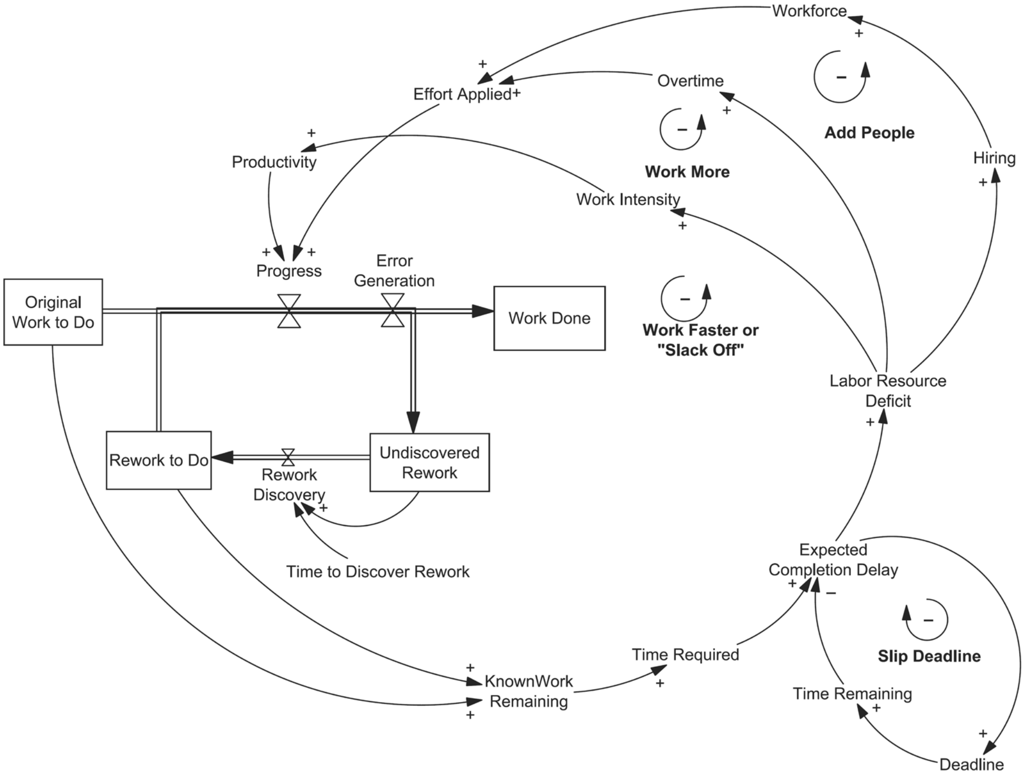


Fig. 7 Some feedback loops commonly incorporated in System Dynamics project models, interacting with the canonical rework cycle at the centre. Reproduced from Lyneis and Ford (2007) with permission of Wiley. Copyright ©2007 John Wiley & Sons, Ltd

concurrently with work is being done. Other models, such as the Task DSM developed by Eppinger et al. (1994), view a DDP as a sequence of tasks in which feedback can occur when an earlier task requires information from a later one. In such cases, an assumption must be made to proceed. The downstream task may eventually provide feedback that the assumption was incorrect, necessitating rework. Whereas the aforementioned models view feedback as a desirable mechanism to absorb uncertainty and guide a process towards its goals, Eppinger et al. (1994) and many others view (late) feedback as an undesirable driver of rework, whose possibility can be reduced and whose scope can be limited by appropriate process organisation.

Quantitative approaches The third and final subcategory concerns mathematical or computational models developed to investigate feedback concepts in design and development, with an emphasis on goal-seeking behaviour. Perhaps the most established approach is System Dynamics modelling. A substantial body of DDP System Dynamics research has focused on the interactions between the development process and the surrounding managerial decision-making, which typically involves feedback loops (Joglekar and Ford 2005). System Dynamics models emphasise that the dynamic complexity of a project arises from multiple intertwined feedback loops, delays, and imperfect information (Forrester 1968). To illustrate, consider the summarising model generated by Lyneis and Ford (2007), shown in Fig. 7. This figure shows the so-called canonical rework cycle at the center, in which work packages in a project are attempted, flaws are built in and then discovered after a delay, leading to rework. Surrounding the rework cycle are examples of typical feedback loops representing managerial control. For example, the rate of progress is dependent on effort applied, which is increased by a feedback mechanism representing an increase in work intensity if the project is observed to fall behind schedule. By developing equations to model the depicted relationships between variables, the project's evolution over time can be studied and the impacts of the various loops and management policies (represented as the governing equations that interrelate the variables) can be analysed. This kind of modelling has been extensively applied to demonstrate the importance of feedback in product development project governance, to investigate different managerial policies that might be applied, and to investigate the tipping points at which particular loops or dynamic behaviours become dominant (Lyneis and Ford 2007). Some limitations are that the equations governing relationships between variables are critical to the simulation outcomes, and the feedback loops themselves must be arranged in an essentially static structure.

One application of System Dynamics modelling that illustrates the importance of feedback in managing multi-team development projects concerns how resource should be allocated to a product development process in response to changing demand that is driven, for example, by unpredictable rework and changes. Joglekar and Ford (2005) point out that the common approach of proportional-feedback based allocation is myopic because it considers only the past without consideration of likely future needs. They develop a control-theoretic model to investigate the benefit of basing resource allocation on a local-linear prediction of future demand, showing that this helps to balance short- and long-term objectives and avoid excessive oscillation in the system dynamics—especially in case of delays within the feedback loop. In another simulation of the large-scale DDP that involves feedback-driven goal-seeking, Yassine et al. (2003) develop a dynamic model studying the situation in which design teams working concurrently on different aspects of a coupled problem must seek periodic feedback from an integrator team. By simulation, they show that the delay between decisions and feedback means that teams must make decisions based on outdated information, causing oscillatory situations in which progress is repeatedly thought to be on schedule before falling behind. Several remedies are suggested by Yassine et al. (2003), including reducing the feedback delay and avoiding overly aggressive design changes in response to feedback information.

A related perspective is that goal-seeking in DDP management aims to control a network of decisions that are interconnected in feedback loops—moving that network towards a goal state by influencing individual nodes. In the general case, Liu et al. (2011) consider how structural properties of complex networks influence their controllability, which can be described as the number of nodes that must be influenced to move the system to a desired state. They find that sparsely connected networks in which nodes have widely varying degrees of connection tend to be difficult to control. Networks of product development tasks, which as previously mentioned have been shown to typically have small-world structures (Braha and Bar-Yam 2007), would fall into this category.

2.4.2 Feedback and learning in DDP management

Considering the middle-rightmost perspective of Fig. 1, one important aspect of managing learning in design and development process management is the handling of testing and the feedback information that it generates. For instance, Thomke (1998) builds on his model discussed in Sect. 2.2.2 to examine the value of information gained from testing, showing that the type of test used at different points in the DDP can significantly affect learning rates. Loch et al. (2001) expand on this concept, showing that testing arrays of design alternatives in parallel may be more effective than a sequential design-build-test cycle in situations where the cost of testing is low and the time required for each test is high—but this also reduces opportunities for feedback-driven learning in the design-build-test cycle. Ward et al. (1995) advocate for parallel testing of many design alternatives early in the DDP to learn about the feasible design region. Here, early efforts to gain feedback on many design concepts, combined with delaying decisions until it is possible to distinguish between concepts, is thought to reduce the need for expensive design rework later.

More generally, learning is important in DDP management because decision-making must evolve to meet the needs of a changing context, and also because of the need to improve over time. A particular challenge for learning in large-scale design and development is that there are interdependencies among decisions made by different people—interconnected feedback loops crossing different parts of an organisation cause decisions to trigger knock-on effects that can confound the initial intentions. Considering this challenge, Senge (1990) writes that people tend to mistakenly believe that “cause and effect are close in time and space”, whereas in reality, the consequences of decisions often occur much later than the actions and can manifest elsewhere in the system. When individual decision-makers do not fully perceive the knock-on effects of their decisions, learning from experience may not be effective (Senge 1990). Senge suggests that a learning organisation should therefore aim to appreciate (and perhaps adjust) the structure of feedback loops impacting on a situation rather than adjusting patterns of behaviour in response to events, and that the way to achieve this is through systems thinking. Especially relevant to this article, one systems thinking method developed to appreciate the structure of feedback loops in a complex situation is causal loop diagramming. Walsh et al. (2020), for instance, use such diagrams to explain how control actions in engineering systems design can lead to unintended consequences due to interacting positive and negative feedback loops.

Argyris (1977) discusses another form of learning in the management context, that he terms double-loop learning. Double loop learning concerns the ability of an organisation to react when a goal being pursued through feedback does not benefit the broader organisation. He notes that various barriers to this kind of learning can exist, for instance, individuals perceiving a problem may not want to contradict goals set by managers. Learning in a management context can also be viewed as the acquisition and maintenance of routines that guide behaviour (Levitt and March 1988; see Schønheyder and Nordby 2018, for a discussion specific to the DDP). This is a feedback-driven learning process, in which routines that are deemed successful in achieving goals gain more prominence than those that do not. Levitt and March (1988) discuss numerous complexities in the effectiveness of such learning. These range from uncertainty and ambiguity, which make it difficult to determine a routine's success, to political issues within an organisation. As a situation becomes more uncertain and complex, codified routines are less useful and feedback-driven, responsive activity becomes more important—as it is in the design and development process.

To summarise, extensive research has been done on learning in management and has established various connections between learning and feedback in this context. This section has summarised some concepts relevant to the present article, in particular establishing that management learning occurs through feedback but at the same time, the complexity caused by interacting feedback loops makes learning from experience difficult. Feedback loops can also cause resistance to organisational learning.

2.4.3 Feedback and emergence in DDP management

The last perspective to be discussed is the bottom-rightmost cell of Fig. 1. The following paragraphs focus on how feedback at the management level contributes to causing and controlling emergent characteristics of a large-scale DDP.

Some researchers in the engineering design community address this perspective by considering how feedback allows a large-scale DDP to adapt itself in response to emerging situations. For example, Pulm (2005) presents a social systems perspective on product development, drawing on concepts of second-order cybernetics and in particular, the ideas of "autopoietic (self-reproducing), integrated (the observer is part of it) systems". A key idea is that the organisational units in a project participate in a social process to adapt product systems under development to one another. Pulm writes that each such unit in a development project constitutes a social subsystem defined by dense recurring communications among the members. Also focusing on large scale projects but with more normative objectives, Naumann and Vajna (2004a, 2004b) develop an approach they call adaptive system management. This approach decomposes product development into 13 subsystems arranged into a hierarchy. The subsystems include the designers, the product lifecycle, and the market environment. Naumann and Vajna (2004a, 2004b) propose that each subsystem can be represented as a causal network of factors. Feedback principles are important in this model as the authors describe that some processes are "intuitive and chaotic" requiring active management, and internal and external disturbances counterbalanced through control loops among the factors. In later work, Vajna et al. (2005) outline their autogenetic design theory, in which they contend that design should be viewed as a feedback cycle of evolutionary operators by analogy to evolution by natural selection—in which design and design processes are evolved in a cycle in response to dynamic selection pressures set by their context. Overall, the papers discussed in this paragraph are highly conceptual; in each case the importance of feedback in handling emergent behaviours of the DDP is made clear, but the respective authors do not demonstrate that their viewpoints can provide concrete insights for improvement.

An important discipline in large-scale design and development is systems engineering, for which a central objective is to avoid undesirable emergent behaviours in a system being developed or intervened with. Some researchers have collected principles from systems theory with a view to inform systems engineering, in particular by better understanding the complex systems under development. In such work feedback processes are recognised to be one of the fundamental building blocks of complex systems, but feedback is usually mentioned only briefly without being the main research focus. Work in this category is usually presented on a more abstract level than the work reviewed above. For instance, Adams et al. (2014) reviews literature and generates 7 axioms supported by 30 propositions thought to be representative of real systems. Feedback is shown as one of the propositions and is implied in many others. Another collection of systems principles in the systems engineering context is presented in Sillitto et al. (2018), while Skyttner (2001) collects an extensive list of general systems principles that could be interpreted in this context.

The final topic to be incorporated in our framework is complex adaptive systems theory. In overview this presents a complex system such as a DDP as being governed by interactions among agents who act according to rules reflecting accumulated experience, while the outcomes of those rules lead to individual learning and potential modifications of the rules (Stacey 1995). The theory posits that the dynamic interactions among learning agents result in a complex intermeshing of positive (reinforcing) and negative (counteracting) feedback loops that occur at multiple levels and timescales, and that this is the mechanism by which emergence occurs (Holland 1992). The negative feedback loops cause goal-seeking behaviour and tend to steady the system dynamics at a certain state, absorbing uncertainties and changes in the system. If the state is desirable the feedback can be viewed as virtuous, or if not, as a vicious loop that reinforces undesired behaviours (Tsoukas and Cunha 2017). In a DDP context negative feedback can be desirable for converging on a solution and for effective project and process management. At the same time, negative feedback loops cause resistance to the change and innovation that are essential features of the DDP context. In contrast, positive feedback loops have amplifying or self-reinforcing effects that create instability and drive changes in system state or configuration following a disturbance. Again, this can be desirable or undesirable depending on context (Tsoukas and

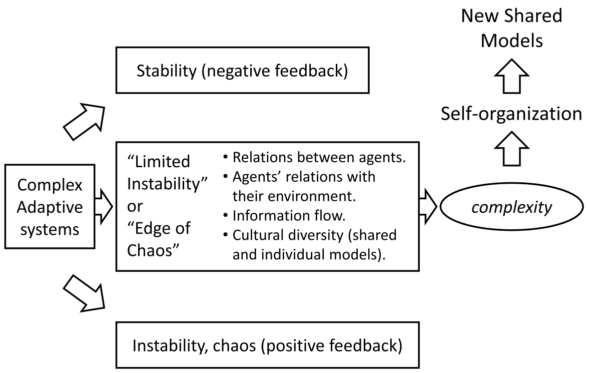

New Shared Models
Stability (negative feedback)
Self-organization
Complex Adaptive systems
"Limited Instability" or "Edge of Chaos"
• Relations between agents.
• Agents' relations with their environment.
• Information flow.
• Cultural diversity (shared and individual models).
complexity
Instability, chaos (positive feedback)

Fig. 8 Complex Adaptive Systems theory aims to explain how creativity, adaptation and emergence result from positive and negative feedback being held in tension, through the interactions of learning agents. Reproduced from Chiva-Gomez (2004) with permission from Elsevier

Cunha 2017). A complex adaptive system is said to exist at the edge of chaos, a situation in which these regimes are in constant tension. At the edge of chaos, some changes may be absorbed while other changes can trigger the system to shift unpredictably to another quasi-stable state or configuration. Applied to the DDP, these complex interactions have been posited to be the root cause of chaotic behaviour and extreme sensitivity to initial conditions (McCarthy et al. 2006), and have been said to help explain how a DDP system can adapt to changing conditions and how creative, innovative outcomes emerge (Chiva-Gomez 2004) (Fig. 8). While these concepts are highly theoretical, some researchers have developed agent-based models of DDP situations to study them from the complex adaptive systems perspective and generate management insights (e.g Garcia 2005; Crowder et al. 2012)—but these models do not emphasise the feedback elements of complex adaptive systems theory, which are of the primary interest here.

Overall, the insights into DDP feedback found in complex adaptive systems theory are closely related to those in second-order cybernetic theory, in that both approaches emphasise the importance of situated learning agents and their idiosyncratic models and goals in emergence of unexpected system behaviours and outcomes. Both are also highly conceptual. But second-order cybernetics focuses on the individual-in-context and how feedback governs their knowing, emphasising the situated construction of understanding and taking a mainly qualitative and conceptual approach. Complex adaptive systems theory, on the other hand, focuses on the implications of feedback-driven situated learning on the behaviour of a dynamic system on a large scale.

2.5 Summary and critique of literature

To recap, the conceptual framework depicted in Fig. 1 draws on selected literature to establish that feedback is essential to the system behaviours of goal seeking, learning and emergence that occur in multiple DDP subsystem types and at different levels of the design and development process. The literature also shows the importance of feedback at different stages of the design process, from early conceptualisation (much of the work discussed in Sects. 2.2 and 2.3 applies to this context) through to the later stages of concurrent engineering projects involving large numbers of participants (the focus of much of the work discussed in Sect. 2.4). Some of the ideas and approaches (e.g. System Dynamics modelling and the Viable Systems Model) have been applied extensively to practice, while other work mainly offers conceptual insight that can help to appreciate the DDP in terms of feedback systems.

A broad range of concepts are represented in the literature. First, researchers addressing goal-seeking behaviour offer insight into how feedback governs the interactions among synthesis and evaluation processes during design, and how it enables a DDP system to converge on or maintain a desired state in an uncertain context. Mainly emphasising self-correcting negative feedback, this perspective lends itself to quantitative analysis of some DDP situations such as resource reallocation in response to changing demand. The goals are sometimes described only implicitly. Work that focuses on feedback as goal-seeking tends to suggest an essentially static feedback system structure, within which information varies over time. Second, researchers addressing learning behaviour offer insight into how knowledge is accumulated and developed through both positive and negative feedback processes, how this evolving knowledge influences decisions in feedback situations, and also how feedback complexity can make learning difficult in a management context. Finally, work focusing on emergent behaviour emphasises the role of feedback in situated construction of knowledge by individual process participants and how this, combined with interactions among feedback processes, can help to explain the mechanisms of creating new designs and the ability of a project to adapt to new information as it is generated, or becomes apparent. In the emergence perspective the possibility of feedback system structures changing over time is allowed for. But research that addresses emergence tends to be highly conceptual in nature, requiring substantial interpretation for a particular situation.

Considering the reviewed work more holistically, it may further be observed that most publications lean towards either (1) a mainly technocentric view which emphasises the ability of a DDP feedback system to cope with uncertainty, imperfect information and dynamic complexity, as exemplified in the work of e.g. Amkreutz (1976), Girard et al.

Table 1 Technocentric and human-centric views on feedback systems in the design and development process

A more technocentric view on DDP feedback	A more human-centric view on DDP feedback
Exemplified in, e.g., Amkreutz (1976), Girard et al. (1999), Joglekar and Ford (2005), Yassine et al. (2003)	Exemplified in, e.g., Holt and Radcliffe (1991), Glanville (2007b), Lurås (2016), Dubberly and Pangaro (2019)
Emphasis on systems, that have human participants	Emphasis on the human as a systems participant
Decision-making is essentially (boundedly) rational	Decision-making is conscious, reflective and idiosyncratic
Objectivist worldview—DDP essentially viewed by analogy to control systems or natural feedback processes; feedback systems are out there to be modelled and improved	Constructivist worldview—feedback systems are constructed by their participants, and in the minds of observers—they are perceived in different ways
Suggests essentially fixed structures of feedback cycles	Suggests fluid and/or subjective feedback structures
Feedback information typically quantitative/objective	Feedback information involves subjective interpretation
Focus on goal-seeking and uncertainty absorption	Focus on how feedback enables knowing and knowledge
Emergence arises from interacting loops, delays, and imperfect information, and is usually undesirable	Emergence arises from coevolving mental models, goals and conversations, and is often desirable
Particular resonance with DDP management processes where the intention is to control, to absorb uncertainty, and/or to converge on a solution	Particular resonance with design activity due to emphasis on situation-dependence and individual experience, and because the constructivist worldview resonates with new concepts, systems and situations being constructed by designing
Model-based approaches ease interpretation for practice	*Discursive approaches offer rich conceptual insights*
...but how well do these perspectives apply to DDP situations that are fluid, creative and emergent?	*...but how can the broad-ranging theory be interpreted to inform design and development practice?*

Note that the example publications do not individually emphasise every concept in the corresponding columns

(1999), Joglekar and Ford (2005) and Yassine et al. (2003); or (2) a more human-centric view in which greater attention is paid to how feedback relates to individuals and their situated construction of knowledge, as offered in e.g. the work of Holt and Radcliffe (1991), Pulm (2005), Glanville (2007b) and Lurås (2016). Differences between these views are summarised in Table 1. On the one hand, human-centric perspectives offer rich insights that have strong resonance with design theory, for instance because they incorporate a constructivist worldview and/or emphasise (co)evolving models as drivers of desirable emergence—but such work also tends to be discursive and highly theoretical. On the other hand, technocentric perspectives are often embodied in more concrete models that are more suitable for practical application, especially to support DDP management. But in treating the DDP by analogy to engineering control systems or as static structures of interconnected feedback loops, the technocentric view deemphasises the inherent fluidity, thoughtful reflection, and social processes of design and development.

One possibility to reconcile these views is to argue that a variety of feedback systems exist or can be perceived in the patterns of interaction that constitute a design and development process. Some of those feedback systems might be relatively well structured and more well suited to description from the technocentric viewpoint, while others might be more fluid and ambiguous, in which case the human-centric view might be more appropriate. But at the same time, many situations in the design and development context could be usefully perceived from either point of view. For instance, situations in mechanical design often involve constructing new understanding (as treated in the human-centric view) but also convergence on quantitative objectives (as treated in the technocentric view). We therefore suggest that both views are relevant to many DDP situations, but observe that much of the literature is oriented towards one side of this spectrum or the other.

When seeking to understand a complex system such as a DDP, a discursive and pluralistic approach is often revealing. At the same time, it would be valuable to synthesise key ideas from both perspectives into a model that could guide their integrated application to practice. Such a model would need to be abstract in nature to encompass the richly ambiguous subject matter, and so that it could be used to frame a range of DDP situations at different system levels.

3 Process model

To create a model offering this synthesis of key concepts, the literature discussed in Sect. 2 was revisited to identify the mechanisms by which feedback is said to enable goal seeking, learning and emergence—i.e. the three behaviours we have focused on in this article, and that are thought to be essential at multiple levels of the design and development process as explained in Sect. 2.1. Insights were synthesised to create a conceptual model.

The new model, termed the Feedback System Function Structure (FS^2) model, distils the essence of DDP feedback situations to be intelligent, reflective process participants

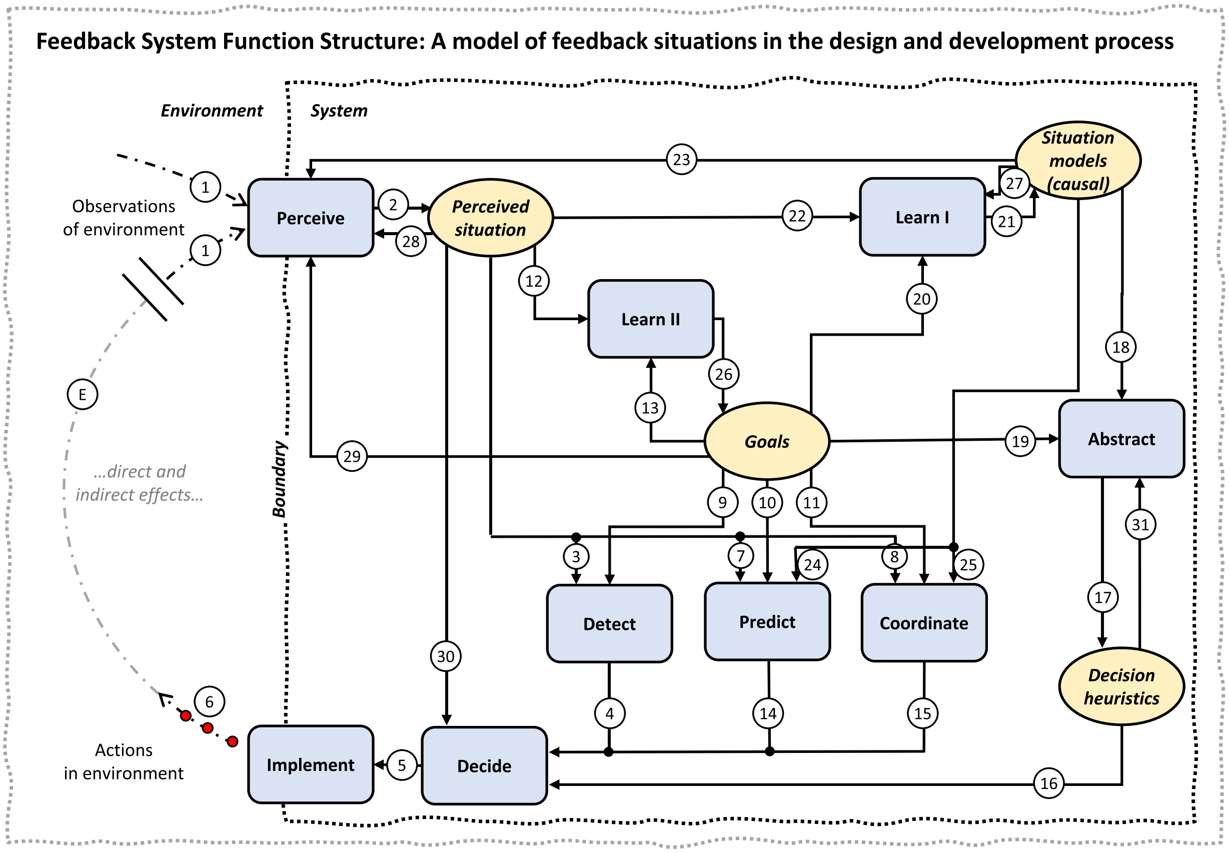


Fig. 9 The Feedback System Function Structure (FS^2) model represents a DDP feedback situation in terms of a system coupled to its environment through feedback loops, and decomposes the system into functions, flows and aspects of memory. Rectangles represent functions. Arrows represent flows of influence, that are each described in Table 2. Ellipses represent aspects of system memory. An unnumbered version of this diagram is provided in the Supplementary Material

(each representing either an individual, or on a more abstract level, a group) purposefully interacting with their environments, that may include other participants. In the model, an individual or group in focus is described as a system, while the broader term *feedback system* encompasses both the system of interest and those aspects of its environment that are closely coupled to it through feedback loops. It is recognised that design and development involves many interacting systems whose actions have different timescales and different scopes of influence. But our model is intended to be generic, so such interactions are not depicted because their structure is generally dependent on the situation at hand—and for many situations can be perceived in different ways. The FS^2 model depicts a system and its environment, decomposes the former into key process functions and aspects of memory involved, and shows how all these elements work together in a cyclic way to enable goal-seeking, learning and emergence in the general case. We propose the model can be applied to frame numerous situations that occur in a design and development project—as long as they can be perceived in terms of feedback processes.

The FS^2 model is depicted in Fig. 9. Two boundaries are shown. The inner boundary defines the system of interest. The outer boundary encloses that system along with the aspects of its environment that are bidirectionally coupled to it, as well as the aspects that may influence the system but are not more than minimally influenced by it. The outer boundary excludes aspects of the environment that no more than minimally influence the system of interest. The model represents a system in terms of functions, depicted as rectangles, that each transform input flows into output flows (these are flows of influence, not sequences of events). Some flows originate or terminate in aspects of system memory, while others occur between two functions. The model adopts the concept that the depicted system is informationally closed—meaningful information exists only within the system given its idiosyncratic frame of reference, and is translated across the boundary into its environment when conversations are held, when emails are sent or are read, when documents, diagrams and CAD models are considered or modified,

Table 2 Flows of influence in the FS^2 model

Flow	Description
1	Aspects of the environment are perceived in the system, e.g., email is read, a CAD model is viewed, etc.
2	Perceptions, manifested as information in the system, influence the perceived situation
3, 9	The perceived situation informs detection of deviations from system goals
7, 10	The perceived situation informs predictions how the environment will evolve with respect to goals
8, 11	The perceived situation informs detection and prediction of deviation from goals in respect to coordination
4, 14, 15	Decisions are made to reduce detected and predicted deviations from goals
16	Decisions are made by applying decision heuristics to reduce deviations from goals
30	The perceived situation is considered when deciding how to reduce deviations from goals
5	Decisions are implemented by actions, e.g., sending email, holding conversations, modifying CAD, etc.
6, E	Actions cause changes in the environment, and some direct or indirect effects may later be perceived
12, 26	The perceived situation, including effects of previous actions, influences the evolution of goals
17, 18	Decision heuristics are abstracted from the system's models of cause and effect in its environment
19	Decision heuristics are formed with respect to the system's goals
20, 21	Learning influences the system's models of cause and effect in its environment and is goal-directed
22	Learning about cause and effect in the environment is influenced by perceptions of that environment
27, 13, 31	Learning involves reflection on existing knowledge
23, 28, 29	The system's models of its environment, and its goals, influence how it perceives that environment
24, 25	The system's models of its environment are used to predict future deviations with respect to its goals

The numbers refer to the labelled flows in Fig. 9

and so on. The translation is accomplished by the functions *Perceive* (involving observations of the environment) and *Implement* (involving actions on the environment), which are depicted as crossing the system boundary to indicate that these are the points at which the system interacts with its environment. Other elements of the model are depicted within the system boundary. This is to indicate that they might not be directly observable, although they are posited by researchers and emphasised in our model to be conceptually-important influences on the behaviour of the feedback system. Note how several cyclic flows of influence exist in Fig. 9. Some of these cycles cross the system boundary, indicating feedback in the system's interactions with its environment. Other cycles occur entirely within the system, indicating reflection and reprocessing of system memory.

Perceiving a DDP feedback situation through the lens of the FS^2 model is in some ways analogous to perceiving an engineering design as a structure of functions that interact through flows of energy, material and information (as described by e.g. Pahl et al. 2007). In particular, in common with a design function structure, our model does not emphasise how the functions, flows and aspects of memory are realised in the physical domain. For instance, it does not depict specific process steps, communication media or people who might be involved.

The following subsections expand upon Fig. 9 to explain each function and aspect of memory, with examples from some common DDP situations. After introducing the model, it will be compared against prior work to show that, although all elements of the model are individually well established in literature, the FS^2 model is the first to synthesise them in context of the design and development process. Some applications of the model to guide structured reflection on design and development processes will then be discussed, and other potential applications and implications will be suggested.

3.1 Functions relating to goal-seeking

Arguably the most fundamental behaviour of intelligent participants in feedback situations is goal-seeking (Pask 1969) which, to recap, enables equilibrium to be maintained and/or objectives to be approached in a context of uncertainty. In the FS^2 model, goal-seeking involves four main functions that occur once aspects of the environment have been observed (via Flow 1 in Fig. 9) and manifested as information within the system (via Flow 2). In particular, the information contributes to the perceived state of the environment, which is processed to *Detect* deviations from the goals (via Flow 3) and/or to *Predict* them in advance (via Flow 7). Adopting the view of O'Donnell and Duffy (2002), the model considers that in a DDP situation at any level, goals relate both to design objectives and management (e.g. coordination) objectives for the task at hand. Additionally, the system participant(s) must *Decide* how to respond to deviations (via Flows 4 and 14). Finally, they must *Implement* decisions (via Flow 5), which means taking actions to influence the environment (via Flow 6). Actions cause direct and indirect effects in the environment (Flow E), some of which are perceived in turn to close the feedback loop via Flow 1. The characteristics of the functions (that are italicised) in this paragraph are elaborated in the next subsections.

3.1.1 Detect function

The *Detect* function is concerned with identifying deviations from system goals, which may subsequently be responded to. As shown in Fig. 9 this is achieved by considering the perceived state of the environment in light of the goals. This function is critical to effective goal-seeking yet is also challenging in the DDP context. In particular, design practitioners are often required to work with information that is of poor quality, and can also be overwhelmed with potentially conflicting information (Wellsandt et al. 2018). Inadequate information leads to nonlinear and complex process behaviours (McCarthy et al. 2006). To give one example, imprecise evaluations of an emerging design might lead to slow convergence towards a solution (Huang and Gu 2006).

3.1.2 Predict function

The *Detect* function discussed in the previous subsection relates to feedback, in which deviation from goals is considered after it occurs. While this is fundamental to goal-seeking behaviour in response to unforeseen events and outcomes, it is in essence reactive and (as is well-known from control theory and as discussed by Simon 1981) will lead to a system's decisions lagging the ideal responses, especially when faced by a trend of growth or decline. In real DDP situations feedback-driven decision-making is complemented by feedforward, represented in the FS^2 model by the *Predict* function. Feedforward refers to situations in which attempts are made to anticipate future evolution of the environment, including in response to potential actions, such that introduction of problems can be avoided and/or mitigating actions can be taken before any undesirable consequences fully manifest. In contrast to *Detect*, which is essentially reactive and for which effectiveness depends on the quality and timeliness of information and the ability to make rapid cycles of adjustment, *Predict* is a more cognitive approach in the sense that its effectiveness is more dependent on the system's ability to understand how its environment may evolve over time and how it may respond to changes. In Fig. 9, this is depicted by Flow 24, which indicates the influence of *Situation models* on the *Predict* function (Situation models are further discussed in Sect. 3.2.1.)

3.1.3 Decide function

The model represents *Decide* as selecting an action intended to drive the feedback system closer to system goals, by considering the detected and predicted deviations (via Flows 4 and 14) in light of participants' understanding of how possible actions might change the environment with respect to the goals (via Flow 16). This understanding is represented as *Decision heuristics* in the FS^2 model (see Sect. 3.2.2 for further discussion). The decision process may involve both rational analysis and reflective judgement, depending on the situation. It may draw on other information about the environment (via Flow 30) in addition to the deviations that are to be responded to. The decision process may involve trade-offs where the system's goals are in conflict, and/or where decision heuristics suggest several possible responses.

According to the so-called principle of requisite knowledge, an optimal decision requires (among other factors) the decision heuristics to have a level of complexity requisite to that of the environment with respect to the goals (Heylighen 1992). In design and development requisite knowledge is not usually possible because individual participants are boundedly rational—they must work with incomplete or imperfect information, have limited time to process that information, and hence do not understand the complete situation in full detail (Simon 1981). Also, the evolutionary nature of design and development means that the relevant knowledge cannot be fully specified in advance—some is generated as the project moves forward, only after issues arise (Pich et al. 2002).

3.1.4 Implement function

In the model, decisions are implemented when the system exerts actions on its environment via Flow 6. For instance, emails might be sent, CAD models might be updated, conversations held, and so on. Actions are objectively observable, although their meanings and the underlying intentions are subjective.

Of relevance to the *Implement* function and its impact on feedback system behaviour, the principle of requisite variety is similar to that of requisite knowledge, but refers to the ability not only to select, but also to carry out an appropriate action (Ashby 1958). Ashby formulated this principle as "Only variety can destroy variety". The essence of his argument (in the terminology of Fig. 9) is that if a system's environment can take on a certain number of states pertinent to the system's goals, those goals can only be ensured if the system can offer the same number of states in response, each tailored to counteract a single undesired situation. But in a DDP context, from any (sub)system's viewpoint the actions that are possible are limited in relation to the possible variety of its environment. The rate at which actions can be completed may also be limited relative to the rate at which decisions can be made. To give some examples: an individual may identify solutions to certain problems but might not have authority or consensus to implement them; or the way a product model has been built up in CAD software may make it very time-consuming to make certain changes to the design.

To summarise, the *Detect*, *Predict*, *Decide* and *Implement* functions work in conjunction to provide the fundamental feedback (and feedforward) loop in our model. The loop is closed when actions on the environment contribute to changes in that environment, and some of the direct and/or indirect effects are later perceived in turn.

3.2 Functions relating to learning

The second group of feedback system functions relates to learning, i.e. the behaviour in which a system and its participants generate knowledge and adapt decision-making in response to a changing environment. These functions address the second row of the framework in Fig. 1. Note that learning also involves the functions discussed in Sect. 3.1.

In the FS^2 model, learning is represented as changes to the memory of the system, in particular to three distinct facets of that memory: *situation models*, *goals*, and *decision heuristics*. The model emphasises that these aspects of memory are each inconsistent and incomplete, because they exist within the system and are updated over time in response to perceived information about its environment—information which itself is limited in quantity, quality and/or timeliness. Learning is conveyed in the model to be a significant driver of dynamic complexity in a DDP feedback system, since it is central to intelligent behaviours and since it causes responses to similar stimuli to differ over time.

3.2.1 Learn I function (relating to situation models)

This subsection first explains the concept of situation models, before moving on to discuss their involvement in learning.

As established in Sects. 2.2.2, 2.3.2 and 2.4.2, each system (i.e. individual or group working towards goals) in a DDP situation maintains knowledge of how their environment is expected to respond to changes and how it is expected to evolve over time. In the FS^2 model, this appreciation is viewed as subjective and unique to each system, and is referred to as the system's *situation models*. Situation models allow reasoning about cause and effect via an understanding of relationships in the environment, in contrast to the *perceived situation* (to be discussed in Sect. 3.3.1), which concerns the perceived current state of the environment. To provide some examples: Situation models may include an understanding of a design being developed, that allows the knock-on effect of proposed changes to be (imperfectly) foreseen. They may include an understanding of design processes that allow the impact of the change on project milestones to be (imperfectly) predicted. They may include an appreciation of the social and organisational context in which design takes place, allowing (imperfect) prediction of how other process participants are likely to react if the change is requested. Situation models may draw on explicit models within the system, e.g. documents and CAD models, but these must be brought to life in the minds of system participants to reason about cause and effect in a particular context. Like any model, situation models are less than the situations they represent and are oriented towards a purpose—in this case, reasoning about cause and effect with respect to the goals of the system. In the FS^2 model, situation models are essential to feedback system behaviour because—as shown in Fig. 9—they are used to predict how the system environment might evolve over time and how it might respond to proposed actions (via Flow 24), and they also indirectly guide decisions via Flows 18-17-16.

Returning to the discussion of learning, the *FS*^{2} model depicts one aspect of learning to be the ongoing modification of situation models in response to observations. This is depicted as the *Learn I* function in Fig. 9. An example of this aspect of learning is the propagation and adjustment of design methods that prove helpful and the archiving of those that do not (a situation discussed by Schønheyder and Nordby 2018). The distinction between this type of learning and goal-seeking can be illustrated by an example in which manufacturing engineers identify a part that is unduly expensive to produce and convey that information to design engineers, who, in turn, change the part design and verify that the issue is resolved. This is an example of goal-seeking, because a deviation from a goal has been detected and then corrected through feedback (via the cycle of Flows 1-2-3-4-5-6-E-1). During learning, in contrast, a connection is made between the detected deviation and the situation model that ultimately guides decisions, and the latter is adjusted through another feedback loop (via the cycle of Flows 1-2-22-21-18-17-16-5-6-E-1). To explain this type of learning in terms of the running example, the design engineer might gain insight into why certain design decisions led to manufacturing difficulties, allowing them to change their decision-making to avoid such difficulties in future. This type of learning can also involve reflection and reframing of existing situation models, which in Fig. 9 is represented by the cycles 27-21-27 and 23-2-22-21-23.

3.2.2 Abstract function (relating to decision heuristics)

As discussed above, situation models represent system participants' appreciation of the system environment and how it might evolve and respond to changes. In practice however, many design and development decisions are not based on complex models and analyses of cause and effect, but on simplified heuristics. Argyris and Schön (1994) use the term theory-in-use to describe heuristic rules of thumb that are used in practice to approach a goal in a given situation. In the FS^2 model, this concept is represented by the *Decision heuristics* in Fig. 9. The *Abstract* function represents the development of decision heuristics from situation models, with respect to the goals.

In design and development practice, it is common that many factors will influence how a situation will respond to changes. A person's understanding of the situation is also likely to contain inconsistencies, especially when a situation is ill-defined or incompletely understood. Accordingly, the Abstract function involves formulating strategies for approaching goals, considering which factors are to be taken into account when making decisions and which can be overlooked, and requires the resolution of conflicts in the understanding of a situation.

3.2.3 Learn II function (relating to system goals)

As previously discussed, DDP feedback systems have purposeful behaviour driven by system *Goals*. Goals are typically related to each system's participation in the functioning of a higher-level system, and are interrelated to form what Albers et al. (2012) term a system of objectives. Our model emphasises that goals are essential to a feedback system's behaviour because deviation from the goals is what drives decisions and, ultimately, actions by the system in its environment. As indicated in Fig. 9, the FS^2 model situates goals internally within a system. This reflects the argument of Pask (1969) that the goals of a human participant in a feedback system are underspecified, open-ended, and consciously managed. This is often described as an important factor in the design and development context—for example design goals have been described as initially ill-defined, requiring reformulation as problem and solution are evolved together (Dorst and Cross 2001; Sim and Duffy 2003). One purpose of design goals is "to motivate activity which in turn will generate new goals" (Simon 1981). To give one example, in the design of a complex product like an aircraft, requirements for each product subsystem are generated and flowed down to the appropriate teams, where they might need to be further detailed. The teams doing the subsystem design work would also provide feedback on whether the goals are achievable, and they might be reformulated if it is later discovered that it is not technically possible to meet them.

The FS^2 model emphasises the role of feedback in developing, expanding and adjusting goals by adopting the concept of double-loop learning (Argyris and Schön 1994). In double-loop learning, a connection is made from the observed effect of previous actions (via Flow 12) to goals (via Flow 26) in light of those and other goals (Flow 13), thereby indirectly all other elements of the model. This may be triggered by new observations or may be a result of reflection. In Fig. 9 this type of learning is represented by including the *Learn II* function.

To conclude the discussion of functions relating to learning (and of the aspects of system memory that are impacted by learning) it can be noted that in complex and ill-structured contexts, as are typical in design and development, the need for learning is greater than in simpler situations but learning is also more difficult (Argyris 1976). A contributing factor is that the high rate of change, high level of ambiguity and need to work with preliminary information in the DDP context means that both single- and double-loop learning must take place in near-real time. Difficulties of disentangling feedback adjustments from changes caused by both modes of learning might be expected (Maier et al. 2012).

3.3 Functions relating to emergence

The final set of feedback system functions incorporated in the model is related to emergent behaviours, thereby addressing the bottom row of the framework in Fig. 1. These functions relate to the role of situated, constructed knowledge and to interactions between agents and among feedback loops. Emergence also involves the functions discussed in Sects. 3.1–3.2.

While goal seeking and learning are desirable for all DDP feedback situations, the desirability of emergence depends on the situation at hand. Where the goal is to generate innovative designs, or for a project to be agile in response to changing context, a degree of emergence may be desirable. In other DDP feedback situations emergent or unplanned outcomes are viewed as undesirable, for instance where the goal is to design a safe system (Taylor 2017) or to ensure project delivery on time and budget (Yassine 2007). A design and development project involves feedback situations of both types.

3.3.1 Perceive function (relating to perceived situation)

To recap, the second-order cybernetics perspectives discussed in Sect. 2.3.3 emphasise the importance of a system's internal models in DDP emergence. In this context models are not viewed as artefacts, such as CAD models or Gantt charts, but are viewed as having been brought to life in the minds of their users to inform decision-making. Such models-brought-to-life exist only within the system and therefore are expressed using the vocabulary of the system and the conceivable actions from the point of view of that system. A model inside a system therefore provides (and also by virtue of being a model) an inaccurate representation of the system environment. All stimuli which can be used to improve that model are transformed or filtered in some way by the system's perceptual apparatus—which is reciprocally dependent on the model via the cycle 2-22-21-23-2 in Figure 9. Participants in such a system can only perceive from within the frame of their models and for this reason have blind spots they cannot see (von Foerster 2003). Sterman (1994) explains this by reference to Kuhn (1970)'s theory that a scientific paradigm suppresses ideas that cannot be easily expressed within it. Second-order cybernetics is therefore associated with radical constructivism (von Glasersfeld 1996), in which the question of how well a system's knowledge (models) represent the real world is said to be undecidable and furthermore not relevant to a cybernetic analysis of the system; from this perspective a good model is one that leads to good decisions with respect to goals at hand (Heylighen and Joslyn 2001). Instead of focusing on the content and fidelity of models used for decision-making, second-order cybernetics offers insights into the feedback process by which the models that people create determine how they construct their perception—and thereby influence the reality—of systems in which they participate (Glanville 2007a). In the context of the DDP, what is being constructed is designers' perceptions of the processes in which they participate and of the design they are creating, and thereby, through their actions and interactions, the processes and designs themselves.

The point of view expressed throughout the previous paragraph is incorporated in the FS^2 model by including the *Perceive* function. In particular the information flows depicted in Fig. 9 convey that system memory not only influences decisions, but also what observations are perceived and perceived to be relevant (via Flows 23 and 28) and how they are translated into information within the system. As noted in Sect. 2.3.3, perception also depends on previous actions in the sense that the system environment yields different information depending on how it is queried. One consequence of this perspective is that interdependencies between perceptual and conceptual apparatus, in particular the impact of one's own frame of reference, are likely significant contributors to the emergence of unexpected outcomes during a DDP.

3.3.2 Coordinate function

The functions discussed so far have emphasised a feedback situation in isolation. But design and development involves a collection of such situations, with many systems working together such that each addresses parts of the overall goal. As already discussed, the more densely connected a situation, the more interacting feedback loops can be found within it (Ramasesh and Browning 2014). We also observe that each DDP participant participates in multiple feedback systems. For example, a designer may participate in one system regarding design performance and another regarding schedule adherence, among others.

Drawing on the discussion of literature in Sects. 2.3.3 and 2.4.3 in particular, the FS^2 model considers that emergent behaviours arise in part from the dynamic interactions between coupled feedback systems. In practice the goals of each system (that, to recap, can be an individual or group participating in the DDP) will be imperfectly aligned. Actions of one system will often be perceived in others only after some time elapses, and sometimes indirectly. What is perceived in one system will differ in some respects from what was intended in another. The situation models, decision heuristics and goals, that are internal and unique to each system, therefore coevolve in complex manner as each system reacts to changes induced by others.

As summarised in Sect. 2.3.3, this coevolution contributes to creativity and desirable emergence in design. On the other hand, the resulting difficulty of predicting how a feedback system will evolve creates non-transparency and challenges in guiding the DDP to a desired state—arguably the essence of complexity in many engineering projects (Maurer 2017). Braha (2020) finds that design problem-solving networks with more interacting feedback loops experience greater problems converging on a solution that satisfies all those loops. Sterman (1994) discusses several reasons why dynamic complexity caused by interacting feedback loops reduces the rate of learning: complex systems are not in equilibrium; actions are irreversible; and the changing of many variables simultaneously confounds attempts to understand cause and effect. Although it is therefore challenging to manage the effects of dynamic complexity, it has been shown that practitioners can learn to make better decisions in the face of dynamic complexity caused by interacting feedback loops, once aware of the issue (Senge and Sterman 1992). In the DDP context, it has also been shown that undesirable dynamic properties arising from interaction between agents seeking feedback can be reduced if designers have good appreciation of how their work relates to the overall design objectives, or if a higher-level feedback system manages the interactions (Yassine et al. 2003).

In the FS^2 model, coordination is represented as an influence on decision-making that is governed by the same feedback and feedforward mechanisms as design decision-making—in other words, by detecting and predicting deviations from goals, where those goals indicate the system's coordination approaches and are subject to evolution based on observation of previous decisions' effects. To ensure that the importance of coordination is emphasised when interpreting the model, it is depicted by including a distinct *Coordinate* function.

Overall, to summarise Sect. 3.3, emergence is presented in our model as arising from dissonance between system memory and the system environment, as well as from the dynamic interactions among coupled feedback systems.

Table 3 Comparison of the FS^2 model to some other feedback-centric models of the design and development process

	Σ	Detct.	Pred.	Decide	Impl.	Learn I	Learn II	Abstract	Perceive	Coord.
Suh (1990)	3	✓		✓	✓					
Weber (2007)	5	✓		✓	✓			✓		✓
Amkreutz (1976)	5	✓		✓	✓		✓			✓
Sim and Duffy (2004)	6	✓	✓			✓	✓	✓		✓
Wynn et al. (2010)	8	✓		✓	✓	✓	✓	✓	✓	✓
Dong (2004)	7	✓		✓	✓	✓	✓		✓	✓
Maier et al. (2012, 2014)	8	✓		✓	✓	✓	✓	✓	✓	✓
Wellsandt et al. (2018)	7	✓		✓	✓	✓	✓		✓	✓
Huang and Gu (2006)	3	✓		✓	✓					
Doumeingts et al. (1996)	6	✓		✓	✓	✓	✓			✓
Merlo and Girard (2004)	6	✓		✓	✓	✓	✓			✓
Wilberg et al. (2015)	6	✓		✓	✓	✓	✓			✓
Elezi (2015)	6	✓		✓	✓	✓	✓			✓
O'Donnell and Duffy (2002)	7	✓		✓	✓	✓	✓		✓	✓
Romero et al. (2008)	4	✓		✓	✓					✓
Joglekar and Ford (2005)	5	✓	✓	✓	✓					✓
Yassine et al. (2003)	4	✓		✓	✓					✓

Publications are shown in the sequence they are discussed in Sect. 2. A tick indicates that a particular function is emphasised in the respective publication, although the perspective on the function may differ from that presented in the FS^2 model

3.4 Summary

The FS^2 model collects some insights into feedback systems from literature and integrates them for the DDP context. By decomposing DDP feedback situations into process functions that contribute to goal-seeking, learning and emergence, the FS^2 model focuses attention on how these interrelated and essential behaviours are realised. The model is intended to apply to multiple DDP subsystem types, including activity, collaboration and management subsystems, and at different levels of the design and development process. The model thereby addresses all nine perspectives depicted in Fig. 1.

We undertook two analyses to clarify the gap addressed by our new model. First, to show that the FS^2 model covers a broader scope of concepts than earlier models of feedback in the DDP, each of its functions was mapped against a selection of literature reviewed in Sect. 2. The reason for focusing on a subset of the reviewed literature is that many of the publications underpinning our model do not decompose a feedback system in terms of functions, aspects of memory and specific flows of influence between them. Therefore, mapping a publication against the constructs of our model would not be meaningful in every case—in particular, for much of the work that discusses feedback in terms of cyclic flows of influence, without emphasising the decision-making processes involved. Such publications were therefore excluded from the analysis, which focused only on work in which functions involved in feedback decision-making can be clearly distinguished. Even in such cases, it was common that some functions in our model were clearly apparent in a publication while others were only obliquely suggested, requiring interpretation as to whether they should be included in the mapping or not. To ensure consistency, a set of specific questions was therefore formulated to capture the essence of each function in our model, then each selected publication was evaluated against those questions. The questions are included in the Supplementary Material and the result of the mapping is shown in Table 3. Overall, although a certain amount of interpretation was required to generate Table 3, it confirms that prior models of feedback in the DDP consider only subsets of the concepts that are integrated in the FS^2 model.

Second, Fig. 10 positions the FS^2 model in relation to some existing approaches on two axes, representing the degree to which a particular approach is either model-based or discussion-based, and the degree to which it is technocentric or human-centric. Returning to our critique of literature in Sect. 2.5, the figure depicts how treatments of feedback in DDP literature tend towards either technocentric or human-centric, with few integrating insights from both perspectives. Furthermore, the approaches found in literature that considers most of the functions incorporated in the FS^2 model are largely discursive in nature, while the FS^2 model embeds these insights into a model with the intention to generate more practical implications and possibilities for application of the theoretical insights. Some of these implications and possibilities are discussed in the next section.

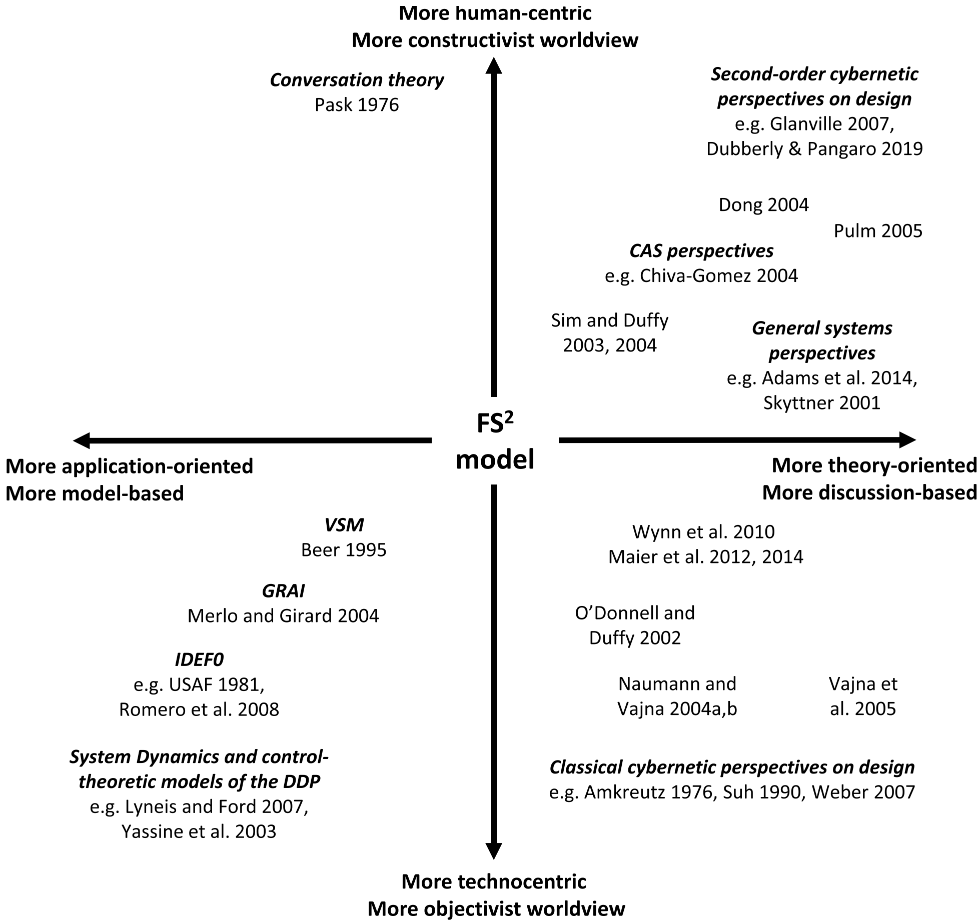

More human-centric
More constructivist worldview
Conversation theory
Pask 1976
Second-order cybernetic perspectives on design
e.g. Glanville 2007, Dubberly & Pangaro 2019
Dong 2004
Pulm 2005
CAS perspectives
e.g. Chiva-Gomez 2004
Sim and Duffy 2003, 2004
General systems perspectives
e.g. Adams et al. 2014, Skyttner 2001
FS²
model
More application-oriented
More model-based
More theory-oriented
More discussion-based
VSM
Beer 1995
GRAI
Merlo and Girard 2004
IDEF0
e.g. USAF 1981, Romero et al. 2008
System Dynamics and control-theoretic models of the DDP
e.g. Lyneis and Ford 2007, Yassine et al. 2003
Wynn et al. 2010
Maier et al. 2012, 2014
O'Donnell and Duffy 2002
Naumann and Vajna 2004a,b
Vajna et al. 2005
Classical cybernetic perspectives on design
e.g. Amkreutz 1976, Suh 1990, Weber 2007
More technocentric
More objectivist worldview

Fig. 10 Positioning the FS^2 model in relation to some other feedback-centric analyses of the design and development process

4 Discussion

The design and development process evidently does not unfold in an uncontrolled and mechanistic way, but is purposefully and consciously influenced by its participants upon consideration of feedback information. To fully appreciate the complex dynamic behaviours of the design and development process, it is therefore important to recognise the role of feedback systems within it. Although feedback is most commonly associated with control or goal-seeking, researchers have also considered it to be essential to emergence and learning—both of which are key behaviours of the design and development process. So, considering the DDP in terms of feedback systems, especially as they relate to the interactions among participants working with idiosyncratic perceptions and goals that are also constructed through feedback, might provide insight into these essential characteristics. But relatively few publications have sought to comprehensively examine feedback in the DDP, in relation to more widely discussed perspectives on that process, such as viewing it in terms of information dependencies or task sequences. We suggest that the topic deserves greater attention.

The high level of abstraction of the concepts discussed in this article may invite criticism on the grounds of practical applicability. In fact the conceptual nature of the work (and much of the reviewed literature) is a necessary and deliberate reflection of the richly ambiguous subject matter. This is because a complex situation like the design and development process involves many individuals with unique perspectives and so cannot be perceived in full by any participant or observer—it needs to be framed to make sense of it. This can be supported by a model that is sufficiently abstract to prompt reflection while also not oversimplifying key issues. The feedback systems perspective discussed in this article and more specifically, the FS^2 model provides such a frame, bringing to light aspects of the DDP related to feedback while submerging and simplifying other aspects. Like any conceptual model, the FS^2 model can support analysis of a situation by prompting balanced consideration of all the issues that are represented in it, helping to ensure that none of those issues are overlooked.

It must be emphasised that because the Feedback System Function Structure was developed to convey conceptual insight it does not offer a 'recipe' or method for making improvements. Like other conceptual models and frameworks, the value of the model hinges on the validity of its concepts, its internal consistency, and ultimately its usefulness for developing insights. Although these characteristics have not yet been formally assessed, we note again that the model was developed by synthesising interrelated concepts that are well established in literature. This provides some confidence in the validity and consistency of the concepts. In an initial investigation of the model's usefulness, we used it to seek insight into situations drawn from our own experience and prior research. Descriptions of three such cases are provided alongside this article. The first case concerns the engineering design process of a machine part. The second case concerns an engineering design process improvement initiative. In the third case, we applied Reich's Principle of Reflexive Practice (Reich 2017) and used the model as a lens to consider the process we had experienced while developing it. The first case is described in the Appendix while the latter two cases are described in the Supplementary Material accompanying this article. As detailed in that text, each application led the authors to new insights into the respective situation, suggesting that the FS^2 model can be useful to guide retrospective reflection on a design and development process.

4.1 Some implications for effective DDP feedback

In technical systems, effective feedback is often defined in terms of goal-seeking, i.e. feedback is deemed effective if it allows a system to converge to stationary or moving objectives as quickly as possible and with minimum oscillation, while absorbing uncertainties. In DDP situations as well, avoiding excessive oscillation is often desirable. For example, this may improve the rate of design convergence and assist learning and coordination by making it easier to identify whether interventions result in improved performance. One factor that causes oscillations and potential instability in a dynamic system is excessive delay in the cause–effect cycle (McCarthy et al. 2006). Research has shown that people often perform poorly in decision situations governed by feedback, and their ability to exert control generally reduces as delays increase (Diehl and Sterman 1995). Long delays in feedback loops may even make it difficult to connect cause and effect at all (Ramasesh and Browning 2014) especially if those loops cut across responsibilities and organisational boundaries (Senge 1990). Another situation in which excessive oscillation is likely, mentioned earlier, is if poor quality of information is used for feedback (Wellsandt et al. 2018). Overall, impediments to the flow of information within a DDP feedback system or between the system and its environment are undesirable when the objective is to approach goals as quickly as possible. Managerial efforts to identify important feedback information and to manage its quality and timeliness are therefore likely to improve DDP performance. The model also indicates that problems are likely if there is a mismatch between the amount and richness of feedback information and the ability to interpret it. Considering the *Perceive* function and its input flows, poor feedback performance can arise when there is too much feedback information but insufficient knowledge or time to extract meaning, or conversely when decisions are based on overinterpretation of limited data. In a large-scale DDP, this reinforces the value of tailoring feedback information to the needs of each decision-making unit.

Another implication is that issues in realising any of the functions or flows in the FS^2 model may cause problems in handling manifested uncertainties, in learning and improvement, and in adapting to new insights generated during the DDP. This suggests that the model might help to pinpoint feedback-related problems and improvement opportunities. For instance, if a particular situation is mapped against Fig. 9 and it is determined that deviations from a desired state cannot be reliably detected or are not reliably transmitted to inform decisions (to provide just two examples), it can be concluded that such deviations will not be reliably absorbed—no matter how well the other functions and flows of the model might be realised. An example is suggested in the Appendix, and similar reasoning is applied in feedback-oriented models for assessing system safety (Leveson 2011), security (Carreras Guzman et al. 2020), for appreciating process modelling interventions (Wynn et al. 2010), and in applications of the Viable Systems Model (e.g. Elezi 2015).

In practical terms several enablers of effective feedback in the DDP context may be suggested. An organisation able to take on new ideas and respond rapidly to new information is likely to derive more benefit from feedback than another that cannot. This might be supported, for example, by agile development processes, a culture of sharing preliminary information, the use of rapid prototyping, and flexibility in the design itself, that might (for example) be enabled by modularity and reconfigurability. On the other hand barriers to effective feedback can include the inverse, such as overly bureaucratic processes and tightly integrated products that include legacy systems which are difficult to change, as well as non-mechanistic issues, such as psychological, cultural, and organisational barriers to providing or responding to feedback information (Busby 1999).

4.2 Conceptual tensions

While much of the literature on feedback in the DDP offers complementary perspectives, tensions may also be identified. One such tension might be perceived between the view that feedback is desirable as in the goal-seeking perspective exemplified in the work of Yassine et al. (2003), and the view that (late) feedback is undesirable as in the perspective discussed by Eppinger et al. (1994). Another point of tension occurs between work that focuses on feedback in goal-seeking and that which focuses on feedback in emergence. In particular, the goal-seeking perspective emphasises the importance of control in guiding a system towards its objectives. According to this perspective an effective feedback system is one that can eliminate uncertainty, and sources of dynamic complexity such as interlocking feedback cycles are seen as undesirable. But on the other hand, research in second-order cybernetics and complex adaptive systems theory emphasises that it is in fact the non-idealities relating to feedback that lead to the important DDP behaviours of creating new designs, and being able to adapt when new opportunities arise. For instance, emergence of new ideas in team designing is related to the difference between participants' mental models (as discussed in Sect. 2.3.3) and an organisation's ability to reconfigure itself in a changing environment is shown to arise from the complexity of interactions among positive and negative feedback loops (as discussed in Sect. 2.4.3). The feedback systems perspective thus may offer insight on why the 'define the goals then control work towards them' approach of classical systems engineering and project management helps to manage projects in the face of internal complexity, but might be less than ideal in situations where goals are difficult to define upfront. This control-based approach is also sometimes associated with resistance to innovation (see Lenfle and Loch 2010).

4.3 Outlook

Several opportunities for further work may be suggested. First, noting that the FS^2 model has only been used on a few cases by the authors to date as discussed in the Appendix and Supplementary Material, we intend to more comprehensively explore its utility. For instance, we hope to investigate whether practitioners and other researchers can identify how the functions in the FS^2 model are realised in particular situations of interest, and whether they believe that perceiving a DDP situation through the frame of the model yields useful insight. The necessary interpretation might conceivably be challenging for those unfamiliar with the underlying theories; perhaps a structured process to guide the model's application would be helpful.

Apart from guiding reflection on DDP situations, other opportunities are evident to apply the FS^2 model. For instance, it could potentially be used to structure computational studies of feedback in the design and development process. To give one example, agent-based simulations might be developed in which each agent is conceptualised as an implementation of the FS^2 model. Considering for instance the focus of the FS^2 model on how feedback systems are governed by their models of the external environment, such simulations might be used (among other possibilities) to investigate the roles of models in the DDP—such as the mechanisms by which models are managed and updated, the impact of misalignment of models in terms of process coordination, and the role of coevolution of mental models in creating novel designs. The FS^2 model might also be used to frame observational or experimental studies of the DDP, perhaps yielding insight into how designers handle feedback situations in practice and how this impacts design outcomes.

Finally, additional conceptual work is also possible to extend and refine the FS^2 model. For example, it would be possible to more explicitly account for interaction among DDP feedback systems, considering issues such as levels of influence and speed of action. Differences might be expected depending on e.g. the scale of the situation and the levels of autonomy of the teams and organisations involved. Considering that the DDP is a temporary project-based activity, emergence of feedback systems as a project evolves and grows or reduces in complexity could also be considered. One challenge would be to concretise conceptual insights far enough to be useful to engineering practice, without compromising the deliberate generality of the FS^2 model. The relevant system-theoretic ideas tend to be discursive and conceptual—very rich in insight, but not straightforward to transform into concrete recommendations for improvement.

5 Concluding remarks

Feedback is one of the main drivers of dynamic behaviours in the design and development process (DDP). This article has collected perspectives on how feedback enables goal-seeking, learning and emergence behaviours, all of which are recognised by researchers to be essential in the DDP at multiple levels. We have argued that an integrated consideration of technocentric and human-centric views on feedback systems is helpful to appreciate goal-seeking, learning and emergence in the design and development context. Technocentric views offer model-based analyses of DDP feedback situations and some quantitative insights that can support process improvements, while human-centric views offer rich qualitative insights and resonate strongly with design situations due to their focus on coevolution, desirable emergence and how knowledge is constructed through

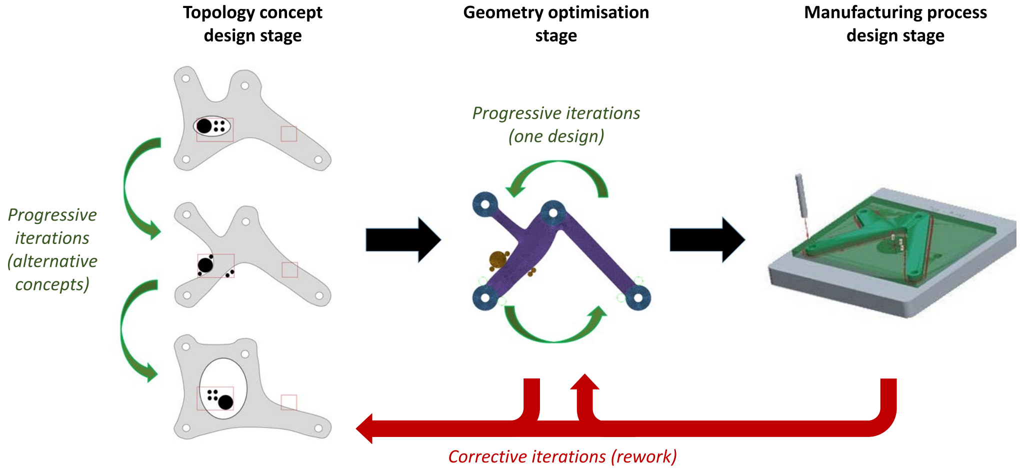

Topology concept design stage
Geometry optimisation stage
Manufacturing process design stage
Progressive iterations (alternative concepts)
Progressive iterations (one design)
Corrective iterations (rework)

Fig. 11 Overview of the mechanical design process discussed in the Appendix. Reproduced from Wynn and Clarkson (2021) with permission of Springer Nature

feedback processes. To synthesise insights from both views, this article has introduced the Feedback System Function Structure, a system-theoretic process model that provides a guide to frame a variety of DDP situations from a feedback systems perspective. Application examples (in the Appendix and Supplementary Material) show how the new model can support structured reflection and can yield insights for research and improvement. We have also discussed some more general implications for improving the effectiveness of feedback in the design and development context. Overall, improvements to DDP feedback systems could help to more effectively handle the uncertainty, novelty and complexity inherent to design and development.

As well as introducing our new model, it is hoped that this article will more generally convey how the feedback systems perspective can provide a useful lens on the design and development process—complementing other perspectives such as the DDP viewed as an iteration process, as a negotiation process, or as a network of relations among actors, tasks and/or information. One advantage of the perspective explored in this article is that it indicates some commonalities across different situations in the design and development process (as depicted in Fig. 1) that are often not examined together in research publications. It can also be noted that DDP feedback systems tend to cut across traditional process boundaries and responsibilities, e.g. connecting individuals with teams and engineering design with process management. In consequence, a feedback systems perspective can help to appreciate the design and development process more holistically as an interconnected and complex dynamic system.

Appendix

This appendix provides an example of interpreting the FS^2 model to generate insight into a specific design situation. To support our claim that the model can be applied to a wide variety of DDP feedback situations and at different scales, more examples are provided in the Supplementary Material accompanying this article.

The application discussed here concerns the mechanical design process for a machine part. It is based on a student project at the University of Auckland, in which the first author has been involved for a number of years. This project has been devised to reflect key features of CAE-intensive mechanical design practice. In particular, students work in pairs to design a machine part to provide specified stiffnesses under two different load cases, avoiding buckling, and minimising both weight of the part and the time required to produce it on a CNC machine. The problem is arranged such that parts with different topologies are possible. Students are advised to approach the task using a three-stage design process, and to expect iteration among the stages. The first stage is to trial different topologies using rapid, low-fidelity Finite Element Analysis (FEA) before selecting the one they think is best, second, refining the selected topology through a series of refinement iterations using higher-fidelity FEA, and third, designing the NC path and predicting machining performance (Fig. 11).

The design process as performed by students is highly iterative and involves all three system behaviours discussed in Section 2.1:

- *Goal-seeking* Part geometry is created in a CAD package and transferred to another package for Finite Element Analysis. Considering the FEA results, the geometry is revised to try to close the gap between the simulated performance and the desired performance. This is repeated iteratively until a satisfactory solution is reached.

- *Learning* As they iterate, students improve their understanding of fundamental solid mechanics by observing how their emerging part's deformation and stress distributions respond to geometric changes. They also develop an understanding of how certain changes are likely to affect the part behaviour.
- *Emergence* In reality, an optimal solution to the task can be generated computationally (although the problem is arranged such that students cannot easily do so). Nevertheless student participants in the project typically submit a range of different topologies, indicating that there are possibilities for emergence. One reason is that the design students settle on is highly dependent on how they search the space of possibilities—the problem is such that it is not possible to exhaustively consider all possibilities in the time available and so, the result depends on the design routes that are followed, the propensity of the group to either refine a single design or keep stepping back to explore the design space, and the trade-off decisions that are made.

An expanded discussion of this design project from the perspective of task sequences and process organisation can be found in Wynn and Clarkson (2021). Here, the same situation is considered from the feedback systems perspective.

Like any system of at least moderate complexity, it is possible to perceive this situation in different ways and draw the boundary to encompass different sets of issues. Here, although the design is done in pairs, it will be framed as a feedback system from the point of view of individual design activity, i.e. focusing on the relationship between the designers and the design information while deemphasising the interactions between the designers themselves. To achieve this framing, each element of the FS^2 model depicted in Fig. 9 was considered in turn and its meaning was identified for the context at hand. This resulted in the following description of the situation as a feedback system:

1. *System boundary* The designers are inside the boundary. The design, as represented in the CAD and FEA tools, is outside the boundary.
2. *Output flow (actions)* The designer makes adjustments to the emerging design models as represented in the CAD tool.
3. *Input flow (observations)* The designer can observe the results of Finite Element Analysis and the results of machining process simulation.
4. *Situation model* The designer appreciates the principles of solid mechanics that govern how the part will deform when loads are applied.
5. *Decision heuristics* The designer applies heuristics for modifying the design to approach their goals considering FEA results. For instance, they might consider that to reduce weight, they could try removing material from regions where Von Mises stress appears low.
6. *Goals* The student is informed that their mark depends on (a) meeting stiffness targets as closely as possible (b) minimising weight (c) minimising machining time. The design performance, and the mark, is calculated as a weighted sum of these objectives. These goals are known in advance and well-defined (which is not the case for many design situations). But even in this case the designer must prioritise the goals and develop a strategy to guide their design activity—for example, to maximise performance they may aim to develop a design that is quick to machine but may be relatively heavy, or an aggressively lightweight design that is slower to machine.
7. *Detect function* Determines the difference between the stiffness of the current design as predicted with FEA and the target stiffness, the difference between the current weight as predicted from the CAD tool and the target weight, and the difference between the machining time predicted by CNC simulation and the target machining time.
8. *Predict function* The designer tries to predict how much further improvement can be achieved through further adjustment of the current topology concept. Eventually, the performance that can be gained by iterating a particular topology will reach a plateau. The designer also tries to predict whether their design performance will be competitive relative to the rest of the class, which will affect their mark and determines whether further improvement iterations are needed.
9. *Decide function* The designer decides how they will adjust the part geometry on this iteration to bring the actual stiffnesses closer to the target stiffnesses. They may decide to make minor adjustments to the current topology, may decide to try a different topology, or may decide to finalise their design. As they become more confident in their evolving decision heuristics, they may choose to make more substantial changes from one iteration to the next, perhaps arriving at a solution more rapidly.
10. *Implement function* The designer adjusts the part geometry in the CAD package with reference to their decision heuristics. (When the geometry is transferred into the FEA software and reanalysed, the feedback cycle will be closed.)

11. *Learn I function (Situation model).* Through iterations, the designer develops insight about the effects of the changes they previously introduced. For example they might realise that the required asymmetry in stiffness strongly depends on the geometry close to the points where load is applied. This indirectly alters the pattern of their future iterations.
12. *Abstract function (Decision heuristics)* The designer modifies their decision heuristics as their understanding of how the design responds to changes (their situation model) improves through learning. For instance in the above example, they might adjust their heuristics to include the importance of adding/removing material near the loading points, in response to detected deviations from stiffness goals.
13. *Learn II function (Goals)* The target values for the three goals will evolve as the designer iterates and learns about what is possible to achieve and what design strategy is likely to yield the best overall performance.
14. *Perceive function* The FEA tool can produce a vast range of results, but the student only considers the ones they know how to interpret. Much relevant information is not perceived, and hence cannot be taken into account while iterating the design. Perception depends on the situation model—for example, if the student is unaware of the concept of Von Mises stress, that information about the evolving design will simply not be perceived and will not be available for consideration.
15. *Coordinate function* Not relevant in this situation, since the way the situation has been represented, there is only one feedback system in play.

Framing the situation in this way helps to appreciate the causes of dynamic complexity in the design process and, considering the focus of this example on a university design project, suggests how students might be guided in learning to produce better designs. Some implications relate to the importance of realising the fundamental feedback functions. For instance, the description presented above suggests that effective design will depend on being able to properly configure the FEA tool to produce accurate simulations of the part behaviour (detect); having a clear idea of what is to be achieved (goals); understanding which of the many results from the FEA tool are relevant (perceive); appreciating how adjustments to the emerging design are likely to change its behaviour (decision heuristics); and so on. While these implications might not be surprising, others are perhaps less apparent prior to the analysis presented here. For example, the above description also highlights the importance of purposefully evolving a desired tradeoff between the three design objectives; the importance of the search process on the quality of the final solution; and the importance of purposefully reflecting on iterations to improve decision heuristics. Overall these observations are different from, and complementary to, the insights earlier gained from a task-based analysis of the same design process, which focus on effective task sequences and decomposition of work among a team (see Wynn and Clarkson 2021, for discussion). Each of these two perspectives reveals certain insights while obscuring others—other perspectives are possible as well. As with the analysis of any complex system, a pluralistic approach is recommended.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00163-022-00386-z.

Acknowledgements The authors gratefully acknowledge the contributions of Mogens Myrup Andreasen, John Clarkson and Thomas Howard to earlier work on this theme, published in Wynn et al. (2010), Maier et al. (2012) and Maier et al. (2014). We also thank the anonymous peer reviewers and Yoram Reich for sharing their insights, all valuable to improve this article. Figure 2 is redrawn from Nam P. Suh. 2.882 System Design and Analysis based on AD and Complexity Theories, Spring 2005, Massachusetts Institute of Technology: MIT OpenCouseWare, https://ocw.mit.edu/. License: Creative Commons BY-NC-SA. Figure 3 is reprinted from Computer-Aided Design, Vol 3, J.H.A.E. Amkreutz, Cybernetic model of the design process, Pages 187–192, Copyright (1976), with permission from Elsevier. Figure 6 is reprinted from Computers in Industry, Vol 55, C. Merlo, Ph. Girard, Information system modelling for engineering design co-ordination, Pages 317–334, Copyright (2004), with permission from Elsevier. Figure 7 is adapted by permission from Springer Nature, Integrated Design and Manufacturing in Mechanical Engineering –98: Proceedings of the 2nd IDMME Conference held in Compiègne, France, 27–29 May 1998 by Jean-Louis Batoz, Patrick Chedmail, Gerard Cognet and Clément Fortin (eds) Copyright Springer Science+Business Media B.V. (1999). Figure 9 is reprinted from Technovation, Vol 24, R. Chiva-Gomez, Repercussions of complex adaptive systems on product design management, Pages 707–711, Copyright (2004), with permission from Elsevier. Fig. 12 is reprinted by permission from Springer Nature: Springer, Research in Engineering Design, Improving the engineering design process by simulating iteration impact with ASM2.0, D.C. Wynn and P.J. Clarkson, ©The Author(s), under exclusive licence to Springer-Verlag London Ltd. part of Springer Nature (2021)

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

References

Adams KM, Hester PT, Bradley JM, Meyers TJ, Keating CB (2014) Systems theory as the foundation for understanding systems. Syst Eng 17(1):112–123

Ahmed S, Wallace KM, Blessing LTM (2003) Understanding the differences between how novice and experienced designers approach design tasks. Res Eng Des 14(1):1–11

Albers A, Ebel B, Lohmeyer Q (2012) Systems of objectives in complex product development. In: Horvàth I, Albers A, Behrendt M, Rusák Z (eds) Proceedings of TMCE 2012, May 7–11, 2012, Karlsruhe, Germany

Amkreutz JHAE (1976) Cybernetic model of the design process. Comput Aided Des 8(3):187–192

Andreasen MM, Hansen CT, Cash P (2015) Conceptual design: interpretations, mindset and models. Springer, Cham

Argyris C (1976) Single-loop and double-loop models in research on decision making. Adm Sci Q 21(3):363–375

Argyris C (1977) Double loop learning in organizations. Harv Bus Rev 55(5):115–125

Argyris C, Schön D (1994) Organizational learning II: theory, method, and practice. Addison-Wesley Publishing Company, Reading

Ashby WR (1958) Requisite variety and its implications for the control of complex systems. Cursos Congr Univ Santiago de Compostela 1(2):83–99

Beer S (1995) Brain of the Firm, 3rd edn. Wiley, London

Braha D (2020) Patterns of ties in problem-solving networks and their dynamic properties. Sci Rep 10(1):1–22

Braha D, Bar-Yam Y (2007) The statistical mechanics of complex product development: empirical and analytical results. Manag Sci 53(7):1127–1145

Braha D, Maimon O (1998) The measurement of a design structural and functional complexity. IEEE Trans Syst Man Cybern Part A Syst Humans 28(4):527–535

Brocklesby J, Cummings S, Davies J (1995) Demystifying the viable systems model as a tool for organizational analysis. Asia-Pac J Oper Res 12(1):65–86

Bucciarelli LL (2002) Between thought and object in engineering design. Des Stud 23(3):219–231

Busby JS (1999) The neglect of feedback in engineering design organisations. Des Stud 19:103–117

Butler DL, Winne PH (1995) Feedback and self-regulated learning: a theoretical synthesis. Rev Educ Res 65(3):245–281

Carreras Guzman NH, Wied M, Kozine I, Lundteigen MA (2020) Conceptualizing the key features of cyber-physical systems in a multi-layered representation for safety and security analysis. Syst Eng 23(2):189–210

Chanron V, Lewis K (2005) A study of convergence in decentralized design processes. Res Eng Des 16(3):133–145

Chiva-Gomez R (2004) Repercussions of complex adaptive systems on product design management. Technovation 24(9):707–711

Christiansen TR (1993) Modeling efficiency and effectiveness of coordination in engineering design teams. PhD thesis, Stanford University

Clark KB, Fujimoto T (1989) Lead time in automobile product development explaining the Japanese advantage. J Eng Tech Manag 6(1):25–58

Colquhoun GJ, Baines RW, Crossley R (1993) A state of the art review of IDEF0. Int J Comput Integr Manuf 6(4):252–264

Council Design (2007) Eleven lessons: managing design in eleven global companies–desk research report. Design Council, London

Crowder RM, Robinson M, Hughes HP, Sim YW (2012) The development of an agent-based modeling framework for simulating engineering team work. IEEE Trans Syst Man Cybern Part A Syst Hum 42(6):1425–1439

Diehl E, Sterman JD (1995) Effects of feedback complexity on dynamic decision making. Organ Behav Hum Decis Process 62(2):198–215

Dong A (2004) Design as a socio-cultural cognitive system. In: Marjanović D (ed) Proceedings of DESIGN 2004, the 8th International Design Conference, Dubrovnik, Croatia, May 18-21, Design Society, pp 1467–1474

Dong A (2005) The latent semantic approach to studying design team communication. Des Stud 26(5):445–461

Dorst K, Cross N (2001) Creativity in the design process: co-evolution of problem-solution. Des Stud 22(5):425–437

Dorst K, Dijkhuis J (1995) Comparing paradigms for describing design activity. Des Stud 16(2):261–274

Doumeingts G, Girard P, Eynard B (1996) GIM: GRAI integrated methodology for product development. In: Huang GQ (ed) Design for X: concurrent engineering imperatives. Springer Science + Business Media, B. V., Dordrecht, pp 153–172

Dubberly H, Pangaro P (2019) Cybernetics and design: conversations for action. In: Fischer T, Herr C (eds) Des Cybern. Springer, Cham, pp 85–99

Eder WE (1998) Design modeling–a design science approach (and why does industry not use it?). J Eng Des 9(4):355–371

Elezi F (2015) Supporting the design of management control systems in engineering companies from management cybernetics perspective. PhD dissertation, Technical University of Munich

Eppinger SD, Whitney DE, Smith RP, Gebala DA (1994) A model-based method for organizing tasks in product development. Res Eng Des 6(1):1–13

Finke RA, Ward TB, Smith SM (1992) Creative cognition: theory, research, and applications. The MIT Press, Cambridge

Fischer T, Herr CM (2019) Design cybernetics: navigating the new. Springer, Cham

Forrester J (1968) Principles of systems. Wright-Allen Press, Cambridge

French MJ (1999) Conceptual design for engineers, 3rd edn. Springer, London

Galbraith JR (1974) Organization design: an information processing view. Interfaces 4(3):28–36

Garcia R (2005) Uses of agent-based modeling in innovation/new product development research. J Prod Innov Manag 22(5):380–398

Geyer F (1995) The challenge of sociocybernetics. Kybernetes 24(4):6–32

Girard P, Doumeingts G (2004a) GRAI-engineering: a method to model, design and run engineering design departments. Int J Comput Integr Manuf 17(8):716–732

Girard P, Doumeingts G (2004b) Modelling the engineering design system to improve performance. Comput Ind Eng 46(1):43–67

Girard P, Eynard B, Doumeingts G (1999) Proposal to control the systems design process: application to manufactured products. In: Batoz J, Chedmail P, Cognet G, Fortin C (eds) Integrated design and manufacturing in mechanical engineering '98. Springer, Dordrecht, pp 537–544

Glanville R (2007a) Designing complexity. Perform Improv Q 20(2):75–96

Glanville R (2007b) Try again. Fail again. Fail better: the cybernetics in design and the design in cybernetics. Kybernetes 36(9/10):1173–1206

Hatchuel A, Weil B (2009) C-K design theory: an advanced formulation. Res Eng Des 19(4):181–192

Hatchuel A, Le Masson P, Weil B (2004) C-K theory in practice: lessons from industrial applications. In: Marjanović D (ed) Proceedings of DESIGN 2004, the 8th International Design Conference, Dubrovnik, Croatia, May 18–21, Design Society, pp 245–258

Helfman Cohen Y, Reich Y, Greenberg S (2014) Biomimetics: structure-function patterns approach. J Mech Des 136(11):111108

Hernandez G, Seepersad CC, Mistree F (2002) Designing for maintenance: a game theoretic approach. Eng Optim 34(6):561–577

Heylighen F (1992) Principles of systems and cybernetics: an evolutionary perspective. In: Trappl R (ed) Cybernetics and Systems ‘92. World Science, Singapore, pp 3–10

Heylighen F, Joslyn (2001) Cybernetics and second-order cybernetics. In: Meyers RA (ed) Encyclopedia of physical science and technology. Academic Press, New York

Holland JH (1992) Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. The MIT Press, Cambridge

Holt J, Radcliffe D (1991) Learning in the organizational and personal design domains. Des Stud 12(3):142–150

Huang HZ, Gu YK (2006) Modeling the product development process as a dynamic system with feedback. Concurr Eng 14(4):283–291

Hubka V, Eder WE (1988) Theory of technical systems: a total concept theory for engineering design. Springer, Berlin

Hubka V, Eder WE (1996) Design science: introduction to the needs, scope and organization of engineering design knowledge. Springer, London

IDEO (2015) The field guide to human-centered design, 1st edn. IDEO.org, San Francisco

Joglekar NR, Ford DN (2005) Product development resource allocation with foresight. Eur J Oper Res 160(1):72–87

Klein M, Sayama H, Faratin P, Bar-Yam Y (2003) The dynamics of collaborative design: insights from complex systems and negotiation research. Concurr Eng 11(3):201–209

Kleinsmann M, Valkenburg R (2008) Barriers and enablers for creating shared understanding in co-design projects. Des Stud 29:369–386

Kleinsmann M, Buijs J, Valkenburg R (2010) Understanding the complexity of knowledge integration in collaborative new product development teams: a case study. J Eng Tech Manag 27(1):20–32

Kroll E (2013) Design theory and conceptual design: contrasting functional decomposition and morphology with parameter analysis. Res Eng Des 24(2):165–183

Kroll E, Koskela L (2016) Explicating concepts in reasoning from function to form by two-step innovative abductions. AI EDAM 30(2):125–137

Kuhn T (1970) The structure of scientific revolutions, 2nd edn. University of Chicago Press, Chicago

Lenfle S, Loch C (2010) Lost roots: how project management came to emphasize control over flexibility and novelty. Calif Manag Rev 53(1):32–55

Leveson N (2011) Engineering a safer world: systems thinking applied to safety. The MIT Press, Cambridge

Levitt B, March J (1988) Organizational learning. Ann Rev Sociol 14:319–338

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

Lhote F, Chazelet P, Dulmet M (1999) The extension of principles of cybernetics towards engineering and manufacturing. Annu Rev Control 23:139–148

Li Y, Roy U, Saltz JS (2019) Towards an integrated process model for new product development with data-driven features (NPD^3). Res Eng Des 30(2):271–289

Liu YY, Slotine JJ, Barabási AL (2011) Controllability of complex networks. Nature 473(7346):167–173

Loch CH, Terwiesch C, Thomke S (2001) Parallel and sequential testing of design alternatives. Manag Sci 47(5):663–678

Luhmann N (1995) Social Systems. Stanford University Press, Stanford

Lurås S (2016) Systems intertwined: a systemic view on the design situation. Des Issues 32(3):30–41

Lyneis JM, Ford DN (2007) System dynamics applied to project management: a survey, assessment, and directions for future research. Syst Dyn Rev 23(2–3):157–189

Maier AM, Wynn DC, Andreasen MM, Clarkson PJ (2012) A cybernetic perspective on methods and process models in collaborative designing. In: Marjanović D, Štorga M, Pavković N, Bojčetić N (eds) Proceedings of DESIGN 2012, the 12th International Design Conference, Dubrovnik, Croatia, May 21–24, Design Society, pp 233–240

Maier AM, Wynn DC, Howard TJ, Andreasen MM (2014) Perceiving design as modelling: a cybernetic systems perspective. In: Chakrabarti A, Blessing LTM (eds) An anthology of theories and models of design: philosophy, approaches and empirical explorations. Springer, London, pp 133–149

Mann D (2001) Laws of system completeness. TRIZ J (May)

March L (1984) The logic of design. In: Cross N (ed) Developments in design methodology. Wiley, Chichester, pp 265–276

Maurer M (2017) Complexity management in engineering design—a primer. Springer Vieweg, Berlin

McCarthy IP, Tsinopoulos C, Allen P, Rose-Anderssen C (2006) New product development as a complex adaptive system of decisions. J Prod Innov Manag 23(5):437–456

Merlo C, Girard P (2004) Information system modelling for engineering design co-ordination. Comput Ind 55(3):317–334

Minneman SL (1991) The social construction of a technical reality: Empirical studies of group engineering practice. PhD thesis, Stanford University

Naumann T, Vajna S (2004a) Adaptive system management. In: Vajna S (ed) IPD 2004: Proceedings of the 5th Workshop on Integrated Product Development, Magdeburg, Germany, 22–24 Sep, Design Society

Naumann T, Vajna S (2004b) Adaptive system management. In: Marjanović D (ed) Proceedings of DESIGN 2004, the 8th International Design Conference, Dubrovnik, Croatia, May 18–21, Design Society

Newell A (1981) The knowledge level: presidential address. AI Mag 2(2):1–20

O’Donnell FJ, Duffy A (2002) Modelling design development performance. Int J Oper Prod Manag 22(11):1198–1221

Pahl G, Beitz W, Feldhusen J, Grote K (2007) Engineering design: a systematic approach, 3rd edn. Springer, London

Pask G (1969) The meaning of cybernetics in the behavioural sciences (The cybernetics of behaviour and cognition; extending the meaning of “goal”). In: Rose J (ed) Progress of cybernetics, vol 1. Gordon and Breach Science Publishers, London, pp 15–44

Pask G (1976) Conversation theory: applications in education and epistemology. Elsevier Scientific Publishing Company, Amsterdam

Pich MT, Loch CH, De Meyer A (2002) On uncertainty, ambiguity, and complexity in project management. Manag Sci 48(8):1008–1023

Pulm U (2005) Product development as a complex social system. In: Samuel A, Lewis W (eds) ICED 05: 15th International Conference on Engineering Design: Engineering design and the global economy, Engineers Australia

Ramasesh RV, Browning TR (2014) A conceptual framework for tackling knowable unknown unknowns in project management. J Oper Manag 32(4):190–204

Reich Y (2008) Preventing breakthroughs from breakdowns. In: Biennial Conference on engineering systems design and analysis. Volume 3: Design; Tribology; Education. Haifa, Israel. July 7–9, 2008, pp 93–97

Reich Y (2017) The principle of reflexive practice. Des Sci. https://doi.org/10.1017/dsj.2017.3

Reich Y, Subrahmanian E (2020) The PSI framework and theory of design. IEEE Trans Eng Manag (Early Access). https://doi.org/10.1109/TEM.2020.2973238

Reichelt K, Lyneis J (1999) The dynamics of project performance: benchmarking the drivers of cost and schedule overrun. Eur Manag J 17(2):135–150

Reymen I (2001) Improving design processes through structured reflection: a domain-independent approach. PhD thesis, Technische Universiteit Eindhoven

Romero F, Company P, Agost MJ, Vila C (2008) Activity modelling in a collaborative ceramic tile design chain: an enhanced IDEF0 approach. Res Eng Des 19(1):1–20

Schön DA (1983) The reflective practitioner. Temple-Smith, London

Schønheyder JF, Nordby K (2018) The use and evolution of design methods in professional design practice. Des Stud 58:36–62

Scott B (2001) Gordon Pask's conversation theory: a domain independent constructivist model of human knowing. Found Sci 6(4):343–360

Scott B (2004) Second-order cybernetics: an historical introduction. Kybernetes 33(9/10):1365–1378

Senge P (1990) The Fifth Discipline: the art and practice of the learning organization. Doubleday/Currency, New York

Senge PM, Sterman JD (1992) Systems thinking and organizational learning: acting locally and thinking globally in the organization of the future. Eur J Oper Res 59(1):137–150

Sillitto H, Griego R, Arnold E, Dori D, Martin J, McKinney D, Godfrey P, Krob D, Jackson S (2018) What do we mean by "system"?-system beliefs and worldviews in the incose community. In: INCOSE International Symposium, Wiley Online Library 28:1190–1206

Sim SK, Duffy AHB (2003) Towards an ontology of generic engineering design activities. Res Eng Des 14(4):200–223

Sim SK, Duffy AHB (2004) Evolving a model of learning in design. Res Eng Des 15(1):40–61

Simon HA (1981) The sciences of the artificial, 2nd edn. The MIT Press, Cambridge

Skyttner L (2001) General systems theory: ideas and applications. World Scientific, Singapore

Stacey RD (1995) The science of complexity: an alternative perspective for strategic change processes. Strateg Manag J 16(6):477–495

Sterman J (1994) Learning in and about complex systems. Syst Dyn Rev 10(2–3):291–330

Subrahmanian E, Reich Y, Konda SL, Dutoit A, Cunningham D, Patrick R, Thomas M, Westerberg AW (1997) The N-Dim approach to creating design support systems. In: Proceedings of the ASME 1997 Design Engineering Technical Conferences. Volume 3: 9th International Design Theory and Methodology Conference. Sacramento, California, USA. September 14–17, 1997

Suh NP (2005) 2.882 System design and analysis based on AD and complexity theories, Spring 2005. Massachusetts Institute of Technology: MIT OpenCouseWare

Suh NP (1990) The principles of design. Oxford University Press, New York

Taylor JR (2017) Automated HAZOP revisited. Process Saf Environ Prot 3:635–651

Thomke S (1998) Managing experimentation in the design of new products. Manag Sci 44(6):743–762

Tsoukas H, Cunha MP (2017) On organizational circularity. In: Smith W, Lewis M, Jarzabkowski P, Langley A (eds) The Oxford handbook of organizational paradox: approaches to plurality, tensions, and contradictions. Oxford University Press, Oxford, pp 393–412

Ullman D (2015) The mechanical design process, 5th edn. McGraw-Hill Education, New York

Umpleby SA (1997) Cybernetics of conceptual systems. Cybern Syst 28(8):635–651

Unger D, Eppinger SD (2011) Improving product development process design: a method for managing information flows, risks, and iterations. J Eng Des 22(10):689–699

USAF (1981) ICAM architecture part II—volume IV—function modeling manual (IDEF0), AFWAL-TR-81-4023. Tech. rep., Materials Laboratory, Air Force Wright Aeronautical Laboratories, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio, USA

Vajna S, Burchardt C (1998) Dynamic development structures of integrated product development. J Eng Des 9(1):3–15

Vajna S, Clement S, Jordan A, Bercsey T (2005) The autogenetic design theory: an evolutionary view of the design process. J Eng Des 16(4):423–440

Valkenburg R, Dorst K (1998) The reflective practice of design teams. Des Stud 19(3):249–271

VDI2221 (1987) VDI 2221: systematic approach to the design of technical systems and products. Verein Deutscher Ingenieure

Vickers G (2010) Insights into appreciation and learning systems. In: Blackmore C (ed) Social learning systems and communities of practice. Springer, London, pp 17–34

von Foerster H (2003) Understanding understanding: essays on cybernetics and cognition. Springer, New York

von Glasersfeld E (1996) Farewell to objectivity. Syst Res 13(3):279–286

Walsh HS, Dong A, Tumer IY, Brat G (2020) Detecting and characterizing archetypes of unintended consequences in engineered systems. In: Proceedings of the ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference IDETC/CIE2020 August 17-19, 2020, Virtual, Online, American Society of Mechanical Engineers

Ward A, Liker J, Cristiano JDS II (1995) The second Toyota paradox: how delaying decisions can make better cars faster. Sloan Manag Rev 36(3):43–61

Weber C (2007) Looking at "DfX" and "product maturity" from the perspective of a new approach to modelling product and product development processes. In: Krause FL (ed) The future of product development. Springer-Verlag, Berlin, Heidelberg

Wellsandt S, Thoben KD, Klein P (2018) Information feedback in product development: analysing practical cases. In: Marjanović D, Štorga M, Škec S, Bojčetić N, Pavković N (eds) DS 92: Proceedings of the DESIGN 2018 15th International Design Conference, Dubrovnik, Croatia, Design Society, pp 1745–1756

Wiener N (1948) Cybernetics: control and communication in the animal and the machine. Wiley, New York

Wilberg J, Tommelein ID, Elezi F, Lindemann U (2015) Supporting the implementation of engineering change management with the viable system model. In: 2015 IEEE International Conference on systems, man, and cybernetics, pp 731–736

Wynn DC, Maier AM, Clarkson PJ (2010) How can PD process modelling be made more useful? an exploration of factors which influence modelling utility. In: Marjanović D, Štorga M, Pavković N, Bojčetić N (eds) Proceedings of DESIGN 2010, the 11th International Design Conference, Dubrovnik, Croatia, May 17–20, Design Society, pp 511–522

Wynn DC, Clarkson PJ (2018) Process models in design and development. Res Eng Des 29(2):161–202

Wynn DC, Clarkson PJ (2021) Improving the engineering design process by simulating iteration impact with ASM2.0. Res Eng Des 32(2):127–156

Wynn DC, Eckert CM (2017) Perspectives on iteration in design and development. Res Eng Des 28(2):153–184

Wynn DC, Grebici K, Clarkson PJ (2011) Modelling the evolution of uncertainty levels during design. Int J Interact Des Manuf 5(3):187–202

Yassine AA (2007) Investigating product development process reliability and robustness using simulation. J Eng Des 18(6):545–561

Yassine A, Joglekar N, Braha D, Eppinger SD, Whitney D (2003) Information hiding in product development: the design churn effect. Res Eng Des 14(3):145–161

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.