Design Science

On the integration of product and process models in engineering design

Claudia M. Eckert[1], David C. Wynn[2], Jakob F. Maier[3], Albert Albers[4], Nikola Bursac[4], Hilario L. Xin Chen[3], P. John Clarkson[3], Kilian Gericke[5], Bartosz Gladysz[4] and Daniel Shapiro[3]

1 *Department of Engineering and Innovation, The Open University, Milton Keynes, MK7 6AA, UK*
2 *Department of Mechanical Engineering, University of Auckland, Auckland 1010, New Zealand*
3 *Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, UK*
4 *IPEK Institute of Product Engineering, KIT Karlsruhe Institute of Technology, Karlsruhe 76131, Germany*
5 *Engineering Design and Methodology Group, University of Luxembourg, Esch-sur-Alzette, L-4362, Luxembourg*

Abstract

Models of products and design processes are key to interacting with engineering designs and managing the processes by which they are developed. In practice, companies maintain networks of many interrelated models which need to be synthesised in the minds of their users when considering issues that cut across them. This article considers how information from product and design process models can be integrated with a view to help manage these complex interrelationships. A framework highlighting key issues surrounding model integration is introduced and terminology for describing these issues is developed. To illustrate the framework and terminology, selected modelling approaches that integrate product and process information are discussed and organised according to their levels and forms of integration. Opportunities for further work to advance integrated modelling in engineering design research and practice are discussed.

Key words: product models, design process models, model integration, organising framework

Received 10 December 2015
Revised 25 July 2016
Accepted 19 January 2017

Corresponding author
D. C. Wynn
d.wynn@auckland.ac.nz

Des. Sci., vol. 3, e3
journals.cambridge.org/dsj
DOI: 10.1017/dsj.2017.2

1. Introduction

Product Development (PD) engineers rarely interact with a design directly. Instead, they usually depend on models to express, analyse and communicate the design. Consequently the PD process (PDP) involves a large and heterogeneous set of models. Such models represent many aspects of design including the emerging product, the process by which it is designed, the contexts in which it will be deployed and against which it may need to be tested. This paper discusses some issues surrounding an evolving class of models – those that aim to integrate product and design process information within a single representation.

Different models in the PDP are generated for different reasons and are expressed in different formats depending on their purpose (Gonnet, Henning & Leone 2007). For example, a solid model might be used to define a product's

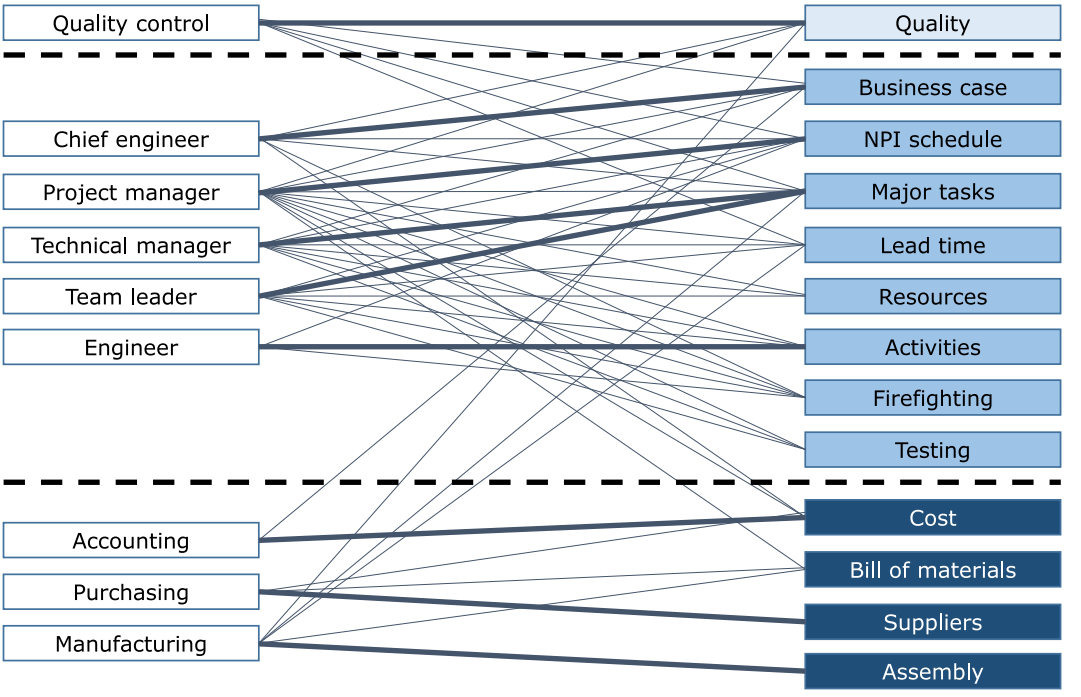

Figure 1. An engineering company uses multiple types of planning document (depicted on the right, named after the main concerns that they include), while stakeholders (depicted on the left) typically need to refer to several plans. Plans are a type of model and this situation is representative of most models used in engineering practice. *Key: Thick solid lines indicate the main plan used by each stakeholder. Thin solid lines indicate additional plans considered. Dashed lines and box shadings indicate the distinction between quality plans, process plans and product plans. Figure adapted from Eckert & Clarkson (2010).*

geometry while a finite element model might be used to perform certain analysis tasks. The same design may be decomposed in different ways depending on the task that needs to be carried out. For example, many products are developed in functional units, such as a car's power train or chassis, but also must be analysed according to whole product properties which can require models on a different level of detail or abstraction. Similarly design processes are considered for different purposes by different stakeholders in an organisation. In consequence such processes are also represented in many different forms, including high-level gateway plans alongside more detailed Gantt charts, flowcharts and to-do lists. Other models such as PLM-related models and financial models are also used to express a PDP and monitor its progress. This was highlighted by Eckert & Clarkson (2010) who identify the multitude of different plan documents used in an organisation they studied (Figure 1). Each type of plan document shown on the right hand side of Figure 1 represents some selective models that place the indicated concerns in context of the overall plan of work.

Figure 1 suggests that while stakeholders may need to consult several planning documents, the plans themselves are not explicitly interconnected. This lack of explicit integration is not limited to planning documents, but also is the case for most other models used in PD – even if those models are created within a single tool or a suite of interacting tools. Of particular interest in this article, product

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

and design process models are often not connected explicitly. Participants in the PDP therefore need to make mental connections between the different models to understand the PDP as a system, which is important to plan, execute and manage it effectively.

1.1. The case for increased model integration

A number of authors have suggested ways that better understanding of the relationship between designs and the processes through which they are generated could support engineering practice. One such situation is when trade-offs must be made between the emerging product and the design process characteristics. This occurs frequently. For example, when deciding how to respond to an engineering change request, a team will need to assess the benefits against anticipated costs of carrying out the change. Assessing change costs involves identifying which parts of the design might be affected – and also involves understanding the redesign processes so that resourcing implications can be considered (Ariyo, Eckert & Clarkson 2009). In this example, it is likely that overlooking some implications on either product or design process will lead to a change that has less overall benefit than originally anticipated. Another situation where product and design process issues must be considered together is when managing the iterative improvement of components or systems. This is because additional iterations might improve design performance, but this benefit must be balanced against the schedule implications (Wynn & Eckert 2017). Again, making such decisions effectively requires trade-offs between product and process considerations, which might in principle be facilitated by integrated models. Research has also suggested that models which integrate product and design process information could support knowledge capture and reuse (e.g., Abramovici & Chasiotis 2002); could provide a basis for AI approaches to design by contextualising design knowledge (e.g., Klein 2000); could assist with conflict management in collaborative processes (e.g., Ouertani & Gzara 2008); and could help to direct the design process considering its dynamic relationship with the emerging design (e.g., Clarkson & Hamilton 2000; Robin, Rose & Girard 2007).

Having noted some situations where integrated representation of product and design process information may be useful, we also suggest the potential value of such models may be increasing. For instance, many companies face continually more challenging PD performance targets such as time, cost and quality, which implies that more issues need to be considered concurrently. At the same time the complexity and sophistication of designs in many domains, such as aerospace, seems to be increasing. Managing increased concurrency and complexity necessitates more information flow between business functions and engineering disciplines during PD. The increased need to coordinate these cross-domain interactions increases the need to consider product and design process together, for example in the scenarios described in the previous paragraph.

1.2. Contribution of this article

Researchers have begun to develop approaches that integrate PD information which is more typically captured in separate models. Analyses published by several researchers have also suggested how so-called integrated models could help with managing interrelationships across information domains in product development.

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

However, there are few publications that review those approaches from the perspective of how models and information domains are, or can be integrated. In the only recent analysis we found, Heisig, Caldwell & Clarkson (2014) review the classes of information element provided by 15 integrated models developed to support knowledge modelling and management. Heisig *et al.* (2014) provide a detailed analysis of the syntax and semantics of the classes used in each of the models, for example comparing different definitions of 'function' in each model. They also compare the various classes found in different integrated models to information needs derived from empirical studies, and rank the models according to their 'coverage' of these needs.

The thorough analysis of information coverage provided by Heisig *et al.* (2014) suggests that certain issues still remain to be clarified. First, there is a need to emphasise the importance of links *between* information elements in achieving model integration. Second, there is a need to consider how information coverage as analysed by Heisig *et al.* (2014) is related to the stated purposes of the models. Third, a clear distinction between the 'language' of a modelling approach and how it is applied in the field may help to express issues relating to model integration in practice. A vocabulary for describing these different issues surrounding model and domain integration has arguably not yet emerged.

The present article aims to contribute on these issues, focusing on the integration of product and design process models. Vocabulary is proposed for describing the issues surrounding model integration, and a categorisation is introduced to organise approaches according to their level and form of integration between product and process domains. Selected key literature on product and process modelling in PD is discussed and classified according to the framework. Areas that deserve further research attention are identified.

1.3. Research method

This article was developed through collaboration between members of the Modelling and Management of Engineering Processes (MMEP) special interest group in the Design Society. We followed a 3-step iterative process, which is explained below.

Step 1

The team started by collecting research publications on approaches to model products and processes in PD, with a focus on approaches that combine these two domains. This was built on previous research effort of the special interest group, which incorporated empirical insights with review of the literature (Heisig *et al.* 2009, 2014). At the same time, the team also searched the literature for a categorisation of models that could be applied or adapted to consider integration of product and process models. A suitable categorisation in the product modelling domain was not identified; however, potentially adaptable work in the process modelling domain was found and analysed. It was decided to build on the categorisation of process models according to purpose by Browning & Ramasesh (2007) on the basis that this categorisation is both comprehensive and well established.

Step 2

In a workshop attended by most of this article's authors, Browning and Ramasesh's categories for process models were considered one by one and

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

extended to also consider the product domain. This generated an extended classification applicable to integrated product and process models. After the workshop the models identified in Step 1 were then classified using the extended categorisation by two of the coauthors and cross-checked by another two. This preliminary categorisation was further developed in a second workshop involving the majority of coauthors. After the second workshop, the whole classification was refined through further discussions informed by additional analysis of the literature.

Step 3

The final step was to validate the proposed terminology and classification. This was approached by systematic search of the literature to ensure that no critical issues or seminal publications had been overlooked. Considering the framework developed in Step 2, the following criteria were developed to determine whether a publication was in scope for our analysis:

(1) *Focus on the engineering design context.* We excluded papers that focused on another domain (e.g., modelling product and design process in software engineering) even if the described approaches might conceivably be applied to engineering design.

(2) *Focus on processes involving multiple human participants in design activity.* We excluded work on design automation, e.g., Potter *et al.* (2003), as well as that focusing on detailed cognitive processes in design, such as Kurakawa (2004). Papers also needed to explicitly focus on design activity, for example work that represents product features alongside manufacturing processes was considered out of scope.

(3) *Potential to model specific situations*. We exclude papers that provide general analysis regarding the link between design processes and the resulting products, such as Demoly *et al.* (2012).

(4) *Coverage of both product and design process domains.* Papers that propose a way to model design tasks, activities or processes also need to explicitly represent product elements such as parameters, components or functions (generic deliverables are not considered to qualify). Likewise product-focused work that does not explicitly involve design tasks are excluded. Also, papers that assume a 1:1 mapping between product and design process domain are not considered integrated here. For instance, we exclude models that represent subsystems in a design and analyse its design process by assuming that each subsystem is developed through a single task (e.g., Maier *et al.* 2014). We also exclude models that assume that each information flow between tasks involves design parameters but do not explicitly represent them (e.g., Chua & Hossain 2012).

A search was undertaken to find publications that meet these criteria but which had not been identified in Steps 1 and 2. This was done by reading all abstracts for articles published from January 2000 until May 2016 in each of the following journals: *Journal of Mechanical Design*, *Research in Engineering Design*, *Journal of Engineering Design*, *Design Science*, *Artificial Intelligence in Engineering Design and Manufacturing* and *Journal of Computing and Information Science in Engineering*. In an effort to avoid oversights each journal was studied independently by two of the authors. In addition, two other authors independently carried out general

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

keyword searches to find additional relevant publications in other venues and prior to 2000.

The results of this search were considered in a third face-to-face workshop attended by most of the authors, to discuss the findings and whether any important areas of research had been initially overlooked. This yielded a number of additional insights, relevant publications and adjustments to the categorisation, all of which were integrated into the final version of this paper. Some models that were originally considered, such as Maier *et al.* (2014) and Chua & Hossain (2012), were also eliminated from the detailed review as the terminology and scope were crystallised.

1.4. Outline of the article

The rest of this article proceeds as follows. Section 2 discusses models and their use in PD and considers how to express the different ways that models can be integrated. Section 3 develops vocabulary for categorising models along three main dimensions: purpose, domain integration and model instance integration. Section 4 analyses the research literature using this vocabulary. Section 5 reflects on the issues revealed and Section 6 concludes.

2. Models

Models are abstractions of reality that are generated for a specific purpose (Frigg 2003). To set the scene, this section discusses some key insights into models and modelling as reported in philosophy of science and product development literature, elaborating the potential value of more integrated models to support design and PD.

2.1. Key points from research into models in the philosophy of science

Throughout the 20th century philosophers of science have increasingly recognised the importance of models in acquiring and organising knowledge. Models are required when it is not practicable to interact directly with a system, for example because of complexity, access limitations, or time and cost implications.

A model can be seen as 'an excerpt of reality' (Stachowiak 1973) or as an expression of intention for reality. As such modellers need to select which aspects of a system they want to express, which can be described as the target system for the model.

Models are often described as 'isomorphic' or 'similar' to their target (Suppe 1977). While a model itself cannot be true or false, the similarity relationship between the model and its target could arguably have a truth value (Giere 2004). Cartwright (2003) however points out that achieving similarity to a target is not the only purpose of models – as George Box famously wrote, 'all models are wrong, but some are useful' (Box & Draper 1987). There are many useful models that are not very similar to their target while other models, which may be more similar, might not be as useful in practice. Models draw attention to one aspect of the target, potentially at the expense of others. Overall, this confirms that the purpose and content of a model are both important when considering whether it is a 'good' or 'sufficient' model.

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

2.2. Models in product development

While models in design share many characteristics of models in science, differences are also apparent. Models of products are typically created as part of the design process to describe the emerging product. Process models are generated to design, communicate, plan and monitor the process. These models do not just describe their targets, they also generate their targets. The utility of models often deteriorates over time, for example if the process changes but a model of it is not updated. Because the effort involved in generating models in design can be considerable, they are often reused for a different purpose than that for which they were originally intended. For example, a process model that was originally used to plan a design process or to make decisions about assigning resources to the process, might later be used to monitor the progress of the process and in the end might remain the only record of the process (Eckert & Clarkson 2010). Often the information about the purpose of a model is not explicit in the model itself, so that users may not be aware if a model they are considering has been repurposed.

Working engineers use a multitude of models to represent both products and processes. Individual designers and design teams can move fluidly between different models, if they are familiar with them (Bucciarelli 1994). However, they can only understand the representations and models generated by others if these are so-called boundary objects that both groups are familiar with (Star 1989). In many engineering contexts, the complexity of the product and the process means that designers heavily rely on models to store and access information about their own design tasks and those of others. In practice this information is typically contained in multiple models, so that people have to mentally synthesise different models in order to answer questions that cross their boundaries.

The relevance of more integrated product and process models that could help address this issue may be rising, as suggested in Section 1. Although the topic is important, the literature review done for this article did not reveal any recent comprehensive overview or examination of the issues surrounding model integration in PD – except for Heisig *et al.* (2014), which is discussed in Section 1.2.

3. Classifying integrated approaches

This paper addresses the research gap outlined above by developing a terminology and categorisation framework that clarifies some key issues surrounding integration of product and process models in PD.

The term 'model' can be used in a variety of ways and the concept of model integration can accordingly imply different things. It is therefore appropriate to consider what is meant by 'model' before moving on to discuss model integration. In particular, the rest of this article differentiates between *model instance*, used here to refer to a specific representation of a target; and *formalism*, denoting the notation or language in which a model instance is expressed. Many formalisms are graphical, e.g., defining the shapes that may be used in a process diagram. A specific framework and formalism applied to a specific target does not uniquely determine the resulting model instance, because this also depends on the modeller's decisions during the modelling process (Gericke, Eckert & Wynn 2016). Therefore, the term *approach* is used here to convey how a formalism is (or is intended to be) applied to develop model instances, and how those model instances are expected to provide benefit. These concepts often appear together

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

Table 1. Definitions of key terms used in this article

Term	Interpretation in the context of this article	Example
Model instance	A representation of a target.	A Binary Task DSM representing a specific design process (e.g., as described by Eppinger *et al.* 1994).
Formalism	A notation or language in which model instances can be expressed.	A Binary Task DSM is a square matrix with ticks in the rows and columns indicating situations where the task in the row requires information from the task in the column (e.g., as described by Eppinger *et al.* 1994).
Approach	How a formalism is (or is intended to be) applied to develop model instances, and how those instances are intended to be used to provide benefit.	A Binary Task DSM can be populated through workshops in which participants are asked to systematically consider each cell one at a time, or through surveys. Algorithms including clustering, sequencing and simulation indicate where inappropriate organisation of tasks may contribute to excessive rework in a project (e.g., as described by Eppinger *et al.* 1994; Browning & Eppinger 2002).
Framework	Guides modellers towards specific formalisms and/or aspects of the product and process that should be modelled.	Kreimeyer & Lindemann (2011) provide a framework comprising set of domains intended to describe the different aspects of product development and possible links between them. DSMs (or MDMs) can be constructed to cover any of the domains and links.

within a research paper and are often bundled together under the term 'model', although we will argue they have different implications for model integration. Finally, the term *framework* is used here to indicate work that guides modellers towards formalisms that might be used and/or aspects of the product and process that might be modelled to achieve an integrated representation. These concepts are summarised in Table 1 and further developed in Sections 3.2 and 3.3.

To develop a terminology and categorisation to organise models according to type and level of integration, the research literature was first searched for existing analyses of product and process modelling work that could yield relevant insight. While no well-established categorisation was found for product models, numerous well-established reviews and analyses of process models do exist (including Smith & Morrow 1999; Wynn & Clarkson 2005; Browning & Ramasesh 2007; Eckert & Clarkson 2010; Eisenbart, Gericke & Blessing 2011; Gericke & Blessing 2012).

Building on insights in these existing publications as well as the authors' own prior research a classification for integrated models was developed. We considered that the way model instances are integrated depends on the purpose for which they are created and combined – as well as the characteristics of the approach, framework and formalism that underlie the modelling activity. This led to the

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

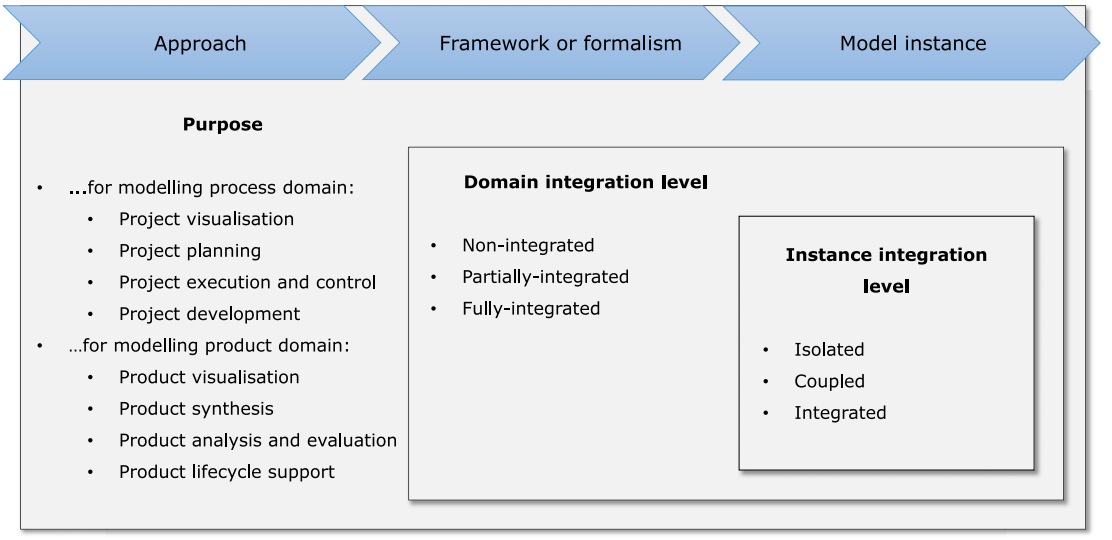

Figure 2. Overview of the categorisation.

following three dimensions, which can help to express issues relating to model integration:

(i) *Purpose of an integrated approach.*
(ii) *Level of integration of a framework or formalism.*
(iii) *Level of integration of model instances.*

The three dimensions and their relationships are depicted in Figure 2 and discussed in the next subsections.

3.1. Purpose of an integrated approach

The first dimension in the categorisation is the purpose for modelling. In this context, purpose is aligned to the modelling approach, although it could arguably also be useful when considering modelling frameworks, formalisms and model instances.

Browning & Ramasesh (2007) analyse process models according to purpose, and we used their work as the starting point to develop a set of purposes for approaches that integrate product and process domains. Browning and Ramasesh's categorisation comprises four major purposes with multiple subcategories. First, process models used for project visualisation illustrate interactions and commitments, provide a common mental model to the workforce and generate views that highlight the most relevant information for certain user groups. Second, process models used for project planning assist in determining necessary activities, in finding efficient process structures, in estimating and improving key process performance metrics and in allocating resources. Third, process models used for project execution and control inform on the progress made and on the best direction to proceed, and also support dynamic replanning of projects. Fourth, process models used for project development capture how design is executed and how it might be executed, thus supporting continuous improvement.

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

This categorisation was extended for the present article to allow organisation of integrated product and process approaches according to purpose. For each of Browning and Ramasesh's subcategories we identified analogous subcategories for modelling the product domain. Each subcategory on the process side was considered and discussed by the authors as explained in Section 1.2 to identify an equivalent subcategory in the product domain. It was not always been possible to find a one-to-one mapping of subcategories across the two domains. Some, e.g., 'Making a commitment' on the process side were accordingly omitted on the product side. For others a broadly equivalent idea was identified, for example 'Structuring the process' can be interpreted as 'Modelling/synthesising product functions, behaviours and/or structures'.

Ultimately, the following categorisation of purposes for modelling product information was developed:

(i) **Product visualisation:** including the visualisation of components, their interfaces, constraints and requirements as well as the generation of views that highlight the most relevant product information for specific user groups. This functionality is, for example, offered by CAD/PLM models and product renderings.

(ii) **Product synthesis:** including the specification and negotiation of requirements as well as the systematic planning and synthesis of product functions, behaviours and/or structures. For example, Sys/ML models are used for this, as are CAD/PLM models.

(iii) **Product analysis and evaluation:** including evaluating the design's adherence to specifications as well as its usability and usefulness. For example, finite element analysis models including loads and boundary conditions are used for this purpose.

(iv) **Product life cycle support:** including support of the design project through documentation and knowledge transfer, engineering decision support, e.g., relating to engineering change management, and support of other business functions, e.g., manufacturing or logistics. For example, Bills of Material and configuration management documents address this purpose.

Table 2 shows the categorisation that was developed to combine the established framework of Browning & Ramasesh (2007) with the extension described above. As mentioned at the start of this section, this categorisation was developed to assist in the analysis of approaches but can also be used to analyse frameworks, formalisms and model instances, noting that multiple purposes can apply in each case.

3.2. Level of integration of frameworks and formalisms

The next subsections elaborate the differences between frameworks and formalisms in the proposed terminology, before considering how to characterise their levels of integration.

3.2.1. Frameworks

Frameworks offer explicit ways of thinking of how products are or should be developed and use this logic to guide users to different formalisms or model instances, clarifying the relationships between them. In this article a framework is

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

Table 2. Categorisation framework for integrated product and design process modelling approaches, extending the framework of Browning & Ramasesh (2007) which is shown in the left column

Purposes for modelling design process domain	Purposes for modelling product domain
Project visualisation	**Product visualisation**
(i) Actions, interactions and commitments	(i) Components, interfaces, constraints, requirements
(ii) Customised views	(ii) Customised views
Project planning	**Product synthesis**
(i) Making commitments	(i) Specification and negotiation of requirements
(ii) Choosing activities	(ii) Modelling/synthesising product functions, behaviours and/or structures
(iii) Structuring the process	
Project execution and control	**Product analysis and evaluation**
(i) Monitoring commitments	(i) Verifying against specifications, i.e., evaluating the product, including maturity
(ii) Assessing progress	(ii) Validating against user needs, i.e., evaluating the product use
(iii) Redirecting	
(iv) Replanning	
Project development	**Product life cycle support**
(i) Continuous improvement	(i) Documentation and knowledge management
(ii) Organisational learning and knowledge management	(ii) Decision support on life cycle issues
(iii) Training	(iii) Support of business functions
(iv) Compliance	

described as integrated if it explicitly considers both product and process domains. Overall, frameworks embody particular views of product development that guide the selection of what to model and in some cases how to represent it. They can empower their users by guiding them to what should be modelled, but might also constrain them to think about PD from a particular perspective.

3.2.2. Formalisms

Modelling formalisms are languages or notations for describing particular aspects of systems. In this article an integrated formalism is defined as a formalism that incorporates both product and process domains.

Formalisms can provide different *levels of abstraction*. For example, Design Structure Matrices (DSMs) (Eppinger *et al.* 1994), Domain Mapping Matrices (DMMs) (Danilovic & Browning 2007), and Multiple Domain Matrices (MDMs) (Lindemann, Maurer & Braun 2009) may be considered abstract formalisms because they allow the modeller choice in how to express heterogeneous information. For example, DSMs are often used to represent elements and

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

relationships in a single domain, e.g., in product, process and organisation DSMs (Browning 2001). Nevertheless, the DSM formalism conceptualises system elements in sufficiently abstract terms that it does not impede mixing domains. In other words, product and process domains can be combined in one DSM model instance, but the modeller has great flexibility in how to do it (for a comprehensive review of DSM-based work, the reader is referred to Browning 2016). The more specific Integrated DEFinition (IDEF) formalisms, on the other hand, provide detailed specifications for what should be put into a model instance and how to organise it (e.g., Mayer *et al.* 1995).

As with frameworks, formalisms provide conceptual categories that guide and constrain the modeller. Formalisms may be explicit or implicit and may be specific or ambiguous to a greater or lesser degree. For instance, an implicit and ambiguous formalism is arguably adopted when an informal flowchart diagram is used to represent a process.

3.2.3. Level of integration of frameworks and formalisms

The integration level of both frameworks and formalisms may be classified according to the different domains they consider. In the categorisation proposed here, a framework or formalism that explicitly incorporates modelling elements in both product domain and design process domain, as well as dependencies within and across these domains, is considered *fully integrated*. Frameworks and formalisms that consider only a partial subset of the possible dependencies but do involve elements in both product and process domains are considered *partially integrated* (this distinction is clarified by examples in Section 4.3). Frameworks and formalisms that consider only elements and dependencies within either the product domain or design process domain are considered *non-integrated*.

As noted earlier, in interpreting these definitions the present article considers that product domain implies a focus on specific design characteristics, and that process domain implies a specific focus on the design process. Therefore, for example, frameworks and formalisms that focus on design processes as collections of tasks along with generic deliverables or generic information flows are considered non-integrated, because deliverables and information flows do not *explicitly* represent design characteristics of the product.

3.3. Level of integration of model instances

Modelling frameworks and formalisms must be deployed in a particular context to develop a model instance, which represents a specific product and/or design process. The level of integration of model instances ultimately depends not only on the approach, framework and formalism that are used, but also on the modeller's decisions regarding how the model instances are developed, maintained and used.

The classification proposed here considers three possible levels of integration of model instances: *Isolated model instances*, being the focused and individually developed models typically found in practice, *Coupled model instances*, which are sets of isolated instances that have been interconnected; and *Integrated model instances*, which apply a coherent logic to describe elements from multiple domains within a single representation. These three categories concern the way the multitude of models used in a company are interrelated. The categories are discussed in the next subsections.

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

3.3.1. Isolated model instances

Isolated model instances represent selected aspects of a product or process. Although recognised to exist within a broader ecosystem of models in a company, an isolated model instance as defined here does not incorporate explicit links to other model instances. Examples include product model instances like CAD models, or process model instances such as Gantt charts or gated process models. While some of these may aim to represent a system in its entirety many are focused on a particular aspect, for example a CAD model may be focused on a single component. Isolated model instances often have a well-understood (although perhaps implicit) scope and content. They may be integrated in the minds of users but are not explicitly interconnected.

3.3.2. Coupled model instances

A coupled model instance comprises a set of isolated model instances with explicit links made between them. For example, CAD models may be coupled using a PDM/PLM system that reads out the model attributes, allowing their relationships to be organised. To give another example, process maps may be coupled using hyperlinks, textual references, or by being stored in a file system that is organised to indicate their relationships. In principle, coupling of model instances may facilitate consideration of dependencies that are not explicit when considering them independently. Model instances can be coupled within and across levels of abstraction or hierarchical detail. For example, a high-level model instance indicating stages in a development process can be hierarchically decomposed into several model instances each representing details of subprocesses, and hyperlinks may be used to indicate the relationships.

3.3.3. Integrated model instances

In comparison to coupled model instances, which link together model instances that could equally be considered independently, integrated model instances are developed with an underlying conceptual integration of the heterogeneous information.

Figure 3 depicts this distinction, noting that there may not always be a clear dividing line. Within a set of coupled model instances, each constituent model instance may be organised, structured and created independently from the others. Elements from one model instance are explicitly linked to other model instances in the coupled set, but without an overall conceptual integration. For example, PLM tools may represent the relationship between model instances that represent different versions of a part design. In contrast, an integrated model instance reflects an attempt to construct a single representation of both product and process domains having a consistent underlying conceptual structure. Some of the structure may be provided by data flow patterns embedded in the chosen modelling formalism, for instance many integrated formalisms require that tasks in the process domain are specified in terms of product elements they operate on and modify. Examples are provided in Section 4.3 and Table 3. Achieving integration also requires deliberation about how to organise a model instance, for example by using consistent nomenclature.

Integration and coupling each have strengths and weaknesses. It may be relatively easy to generate and maintain model instances that are coupled, because the individual model instances can still be managed and updated somewhat

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

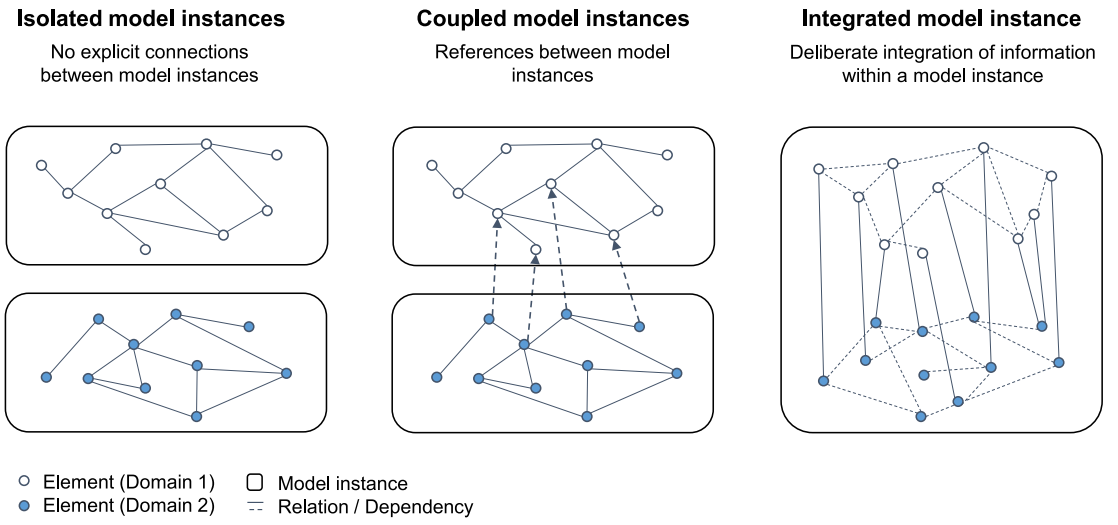

Figure 3. Model instances may be isolated, coupled by explicit cross references, or integrated with an underlying conceptual structure.

independently. However, it might not always be apparent where some models in a coupled set have been modified, or where connections might exist that are not expressed. Therefore, although such models may be easier to maintain than integrated model instances, it may be difficult to ensure consistency. The readers of coupled model instances need overview of the content and must generate their own synthesis. In contrast integrated model instances can facilitate this overview, because the integration is done when constructing the model. However, they may be more difficult to develop, and are likely to require more care and effort to maintain integration between all aspects of the content as the model and the target system evolve.

Bringing different model instances together either by coupling or by integration increases the amount of information that needs to be considered. This poses challenges in terms of usability and data visualisation. However, it might in principle help to reason about trade-offs, identify potential problems and assess the impact of changes. Overall, combining and linking information in either a coupled or integrated model may facilitate reasoning about the interdependencies in complex systems and, in principle, may help teams to develop a shared understanding.

The distinction between isolated, coupled and integrated model instances reflects the approach taken to integrate information. Another issue relating to the integration of model instances is coverage of the target system. For example, a model instance may be integrated as defined above, while not providing equal coverage of product and design process domains.

3.4. Summary

To recap, the categorisation developed in this section is intended to help analyse PD modelling research considering its integration of product and design process information. The categorisation comprises three dimensions: (1) *purpose of the*

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

Table 3. Categorisation by integration level of selected formalisms. See also Table 4

	Elements in product domain	Elements in design process domain	Links within product domain	Links within design process domain	Links across product and process domains	Integration (Partial/Full)
Examples of integrated formalisms introduced in the context of approaches that exploit integrated product–process information						
Signposting (Clarkson & Hamilton 2000)	Design parameters and performance parameters	Tasks	*N/A*	Tasks organised into a hierarchy	All tasks specified in terms of input and output parameters, and associated confidence levels	P
DEPNET (Ouertani & Gzara 2008)	Design specifications (also called technical data)	Activities	Links between 'redundant' product specifications, and links indicating constraints between specifications	*N/A*	Activities specified in terms of input and output technical data	P
APDP (Lévárdy & Browning 2009)	Technical performance Measures	Activities and activity modes (e.g., different ways of performing a test)	*N/A*	Each activity made up of multiple activity modes; Information flows exist between activities	Activity modes affect and depend on technical performance measures	P

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

Table 3. (continued)

ISF (Ahmad, Wynn & Clarkson 2013)	Functions, components, parameters	Tasks	Function–function links, function–component links, Component–component links, component–parameter links	*N/A*	Tasks specified in terms of input and output parameters	P
Examples of integrated formalisms introduced for general knowledge representation						
IDEF3 (Mayer *et al.* 1995)	Object states	Units of behaviour (i.e., actions)	Transition arcs between different object states	Precedence links and junctions (i.e., logic operators) between units of behaviour	Objects are transformed from one object state to another through units of behaviour	F
Design History System (Shah *et al.* 1996)	Parts, attributes, versions, constraints, functional units, assemblies	Processes, tasks, atomic design steps (decomposing, analysing etc.), rationale	Assembly relations, version configurations	Design step hierarchy of abstraction, temporal relations between design steps (sequence, parallelism, iteration)	What product data is used as input, is created, or is modified by a design step	F

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

Table 3. (continued)

SHARED (Gorti *et al.* 1998)	Artefacts, functions, forms (e.g., geometry, material), behaviours	Design operators, design decisions	Various links amongst product domain elements, including hierarchical decompositions	Sequence relationships	Links from design operators to product elements	F
Engineering History Base (Taura & Kubota 1999)	Parts, attributes, constraints, alternatives	Process units, actions	Parts organised into a hierarchy, parts have attributes	Process units have sequence	Product structure used to organise process units; process units set and inherit constraints, process units have actions, actions operate on attributes	F
Design Roadmap (Park & Cutkosky 1999)	Features – can be data (e.g., a number), aggregate properties (e.g., geometry) or artefacts (e.g., drawings)	Tasks, collection nodes (AND/OR input logic gates), branch nodes	Constraints between features, part of relations between features	Tasks have abstraction links above and below; tasks have precedence links to other tasks	Tasks have relations to features: input/output, feedback (F not needed for task, but task is affected if F changes), side effects (if T is executed F is affected)	F

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

Table 3. (continued)

MOKA (Klein 2000; Arndt & Klein 2002)	Functions, Behaviour, Structure, component, assembly etc.	Tasks (analysis and synthesis)	Various links in product domain including hierarchy	Hierarchy of tasks	Process model connected to product model by relations such as goals, requirements, etc.	F
DMS (Whitfield *et al.* 2002)	Product data like design parameters, goals, data files, matrices etc.	Design activities	Product data linked via constraint-based goals	Precedence of activities determined by a DSM	Activities require input to be undertaken and produce output product data	F
Topological structures (Braha & Reich 2003)	Functional specifications, structural descriptions	Closure (design synthesis) and neighbourhood procedures (design analysis)	Structural descriptions fulfil functional specifications	*N*/*A*	Closure and neighbourhood procedures refine functional specifications and structural descriptions	P
Connectivity Map (Yassine *et al.* 2003)	Functions and components	Design tasks	Hierarchy of functions, hierarchy of components	*N*/*A*	Design tasks mapped to functions; each task–function connection is specified by one or more components	P

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

Table 3. (continued)

GRAI (Merlo & Girard 2004)	Function, technological entity, boundary entity	Design activities, Coordination activities	Functions	*N/A*	Activities operate on product domain elements	P
Baxter *et al.* (2007)	Parameters, features (collections of parameters, not geometric features)	Tasks, supporting knowledge (as documentation)	Parameters grouped into sets, sets associated with features; each parameter may link to several features	Tasks linked to supporting knowledge	Tasks operate on features	F
IPPOP (Robin *et al.* 2007)	Product data (Function, Behaviour, Component, Interface, Value etc.) having maturity and status	Activities, having objectives, constraints, triggers; milestones; and iteration mechanisms	Composition links among product data elements	Transitions between activities	Activities operate on product data elements	F
OPM applied to SE (Sharon, de Weck & Dori 2013)	Objects – used to represent 'informatical objects, which are mostly project deliverables'	Processes	Object can 'consist of' other objects or be an 'attribute of' another object	Process can 'consist of' other processes	Instrument links (process requires object) and consumption links (process yields object)	F

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

Table 3. (continued)

Pro2Kon (Abramovici & Chasiotis 2002; Chasiotis 2007)	Components, design models, calculation models, calculation inputs and results	Design steps and methods, calculation steps and methods	Hierarchy of product component 'consists of' relationships, components also 'described by' design models	Processes comprised of process steps, process steps use methods	Process steps specified in terms of product geometry/calculation models used as inputs/outputs	F
FBS-PPRE (Bernard, Labrousse & Perry 2006)	Functions, behaviours, structures, objects, assemblies, states, etc.	Activities, logical flow gateways	Objects decomposed into subobjects; objects represented by representations, etc.	Information flows between tasks	Activities linked to objects, specifications, etc.	F
ASM (Wynn, Eckert & Clarkson 2006)	Parameters (can represent design parameters, and other information that changes during the process)	Tasks, deliverables, process variables	Parameters may be organised into a hierarchy	Tasks may be organised in a hierarchy of subprocesses, tasks may affect and depend on process variables	Tasks specified in terms of input and output parameters	F
CoMoDe (Gonnet *et al.* 2007)	Design objects (models of the artefact, specifications, requirements) and their versions	Activities (analysis, synthesis, evaluation types) and basic operations (design object transformation)	Relationships between design objects, versions of design objects	Activity may be decomposed into subactivities and ultimately into basic operations	Activities generate, delete, modify and/or use design objects	F

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

Table 3. (continued)

Integrated diagrammatic model (Grebici, Wynn & Clarkson 2009)	Product structure (components, assemblies etc.), feature and attribute, material, physical effects and phenomena, product issues, representations	Tasks, decisions, processes, rationale	Any links possible (example: features linked to components)	Any links possible (example: tasks linked to rationale)	Any links possible (example: tasks input/output)	F
DRed 2.0 (Aurisicchio & Bracewell 2013)	Functions, design requirements, criteria, blocks (used to represent components), externally generated pictures such as engineering drawings, and external files	Tasks, questions, answers, rationale	Relationships between functions, between functions and blocks, between blocks	Hierarchical relationships between tasks, information flow between tasks	Process domain activities can be linked to product domain elements (e.g., questions linked to engineering drawings)	F

integrated approach, which focuses on how models are intended to be used to benefit practice; (2) *level of integration of frameworks and formalisms*, which focuses on the interconnection between product and design process domains and (3) *level of integration of model instances*, which concerns how the multitude of fragmentary models used in companies are interrelated. This categorisation is summarised in Figure 2, which also highlights key relationships between the three dimensions. In particular, the purpose(s) of an approach may influence the levels of integration that are necessary or appropriate in a framework, formalism or model instance. Similarly, the framework or formalism will influence the level of integration of model instances. For example, the work of Aurisicchio & Bracewell (2013) could be seen as a framework and formalism that explicitly recommends and facilitates coupling of model instances, while in some other formalisms any coupling must be entirely managed by the modeller.

4. Applying the categorisation to organise the literature

This section provides an overview of some well-established research work that helps to model both product and process information, organised according to the categorisation summarised in Figure 2. The review process was organised as explained in Section 1.3. We focused on publications that explicitly consider both product and design process domains, and on key contributions – as such, our bibliography does not constitute an exhaustively complete list of such work.

Four main groupings of publications were identified as reflected in the categorisation: frameworks; abstract formalisms; partially integrated specific formalisms; and fully integrated specific formalisms. Examples from each of these categories are discussed in the next subsections. While examples from all four categories are given, greatest attention is given to specific formalisms because these arguably provide the most concrete support relating to integration of product and process information in an industry context, which was the original motivation for this article.

4.1. Examples of frameworks

A number of integrated frameworks were identified and key examples are described briefly here. The integrated Product engineering Model (iPeM) (Albers & Braun 2011) defines ten generic activities of a product development process e.g., planning the project, identifying the market opportunities, requirements and evaluation of the product in service. These activities are associated with the set of objectives or requirements and the resulting product description. More detailed product and process descriptions can be associated with each step. The IDEF family of modelling notations shows how to model different views on a product and process (Mayer *et al.* 1995). PSI combines models of the socio-economic and institutional context with product and activity and knowledge models (Reich & Subrahmanian 2015). Architecture frameworks such as the DoD Architecture Framework (DoDAF 2010) indicate the different views that need to be documented and/or integrated on an engineering project. Some frameworks specify detailed formalisms, as IDEF does, while others like PSI do not.

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

4.2. Examples of abstract formalisms

The only examples we found of commonly used abstract formalisms that do not stipulate a categorisation of elements are the Domain Mapping Matrix (DMM) and the Multiple Domain Matrix (MDM). These can be applied to model and analyse the structure of systems that comprise multiple domains (Danilovic & Browning 2007; Lindemann *et al.* 2009).

A DMM is a rectangular matrix similar in principle to a DSM. However, while DSMs can be used to represent the dependency structure between the components of a product or the tasks in a process, as explained earlier, DMMs are used to explicitly map dependencies between elements of different domains (e.g., between tasks and components, although many other mappings are possible). DSMs and DMMs can be combined in an MDM to represent a system's structure across multiple domains. An MDM is essentially a Design Structure Matrix (DSM) that comprises detailed DSMs of single domains along its diagonal with DMMs that map domain pairs outside its diagonal. This allows direct integration of models of the product and design process domain by modelling the dependencies between the elements within and across the domains.

MDMs are often used as the basis of approaches that analyse the structure of systems with multiple domains, e.g., activities, components, requirements and design teams. For example, a number of authors use a component DSM and design process DSM, combined with a mapping matrix that shows which tasks contribute to the design of which components, as the basis of simulations to explore the impact of product changes on the design process (e.g., Gärtner *et al.* 2009; Ahmad, Wynn & Clarkson 2010).

4.3. Examples of specific formalisms

To recap, specific formalisms are not developed in isolation but appear in research publications alongside (or embedded within) approaches, i.e., explanations or hints towards their intended applications.

Some such publications focus on the specification of an integrated formalism that aims to allow users to represent all relevant dependencies amongst domains in a development process (e.g., Abramovici & Chasiotis 2002; Bernard *et al.* 2006; Gonnet *et al.* 2007; Robin *et al.* 2007; Aurisicchio & Bracewell 2013). The aforementioned formalisms are graphical in nature, defining different elements and connections that the authors propose are useful to describe product and design process information in an integrated way. Other authors concentrate on describing a support tool that exploits an integrated perspective on product and design process (e.g., Clarkson & Hamilton 2000; Ouertani & Gzara 2008; Lévárdy & Browning 2009; Ahmad *et al.* 2013). The latter publications do propose integrated formalisms, but those formalisms are not claimed to comprehensively capture the possible elements and links – they are each tailored to the application at hand. They demonstrate how insight can be gained by considering product and process information in an integrated way.

To illustrate the categorisation of formalisms by level of integration, selected specific formalisms were analysed in detail according to the scheme described in Section 3.2.3. The result is shown in Table 3. Overall the analysis of integrated specific formalisms highlights key commonalities about how the approaches integrate product and process information. First, links within either product

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

or design process domains often involve hierarchical organisation of elements. Second, links in the process domain always involve information flows between activities while links from process to product domain always involve specifying the inputs and outputs of tasks. The approaches differ in terms of the particular elements they allow to be described in each domain, as well as other domains (such as rationale) that they consider.

An analysis of the same publications according to purpose of the approaches they describe was also undertaken. The result is summarised in Table 4. Key approaches and issues are discussed in the following paragraphs.

Firstly, to give some examples of approaches that show how integrated information might be exploited, Signposting (Clarkson & Hamilton 2000) guides the selection of activities at each step in the design process based on the designer's confidence in the current state of design parameters. Signposting was enhanced in later work by Melo (2002), O'Donovan, Eckert & Clarkson (2004) and Flanagan (2006) to support design planning. In this approach the advantage of integrated modelling is to capture how the process unfolds based on the current state of the design. This is shown to be more realistic than modelling design as a predetermined sequence of activity. Other authors use integrated models to support engineering change management. For example, DEPNET (product specification DEPendencies NETwork identification and qualification) interlinks product data and design activities (Ouertani & Gzara 2008). The DEPNET database keeps track of the ongoing design process and is used to generate a dependency network of produced product data. If a product datum is changed the representation allows the dependency network to be filtered, to locate likely downstream impacts. The Information Structure Framework (ISF) developed by Ahmad *et al.* (2013) similarly aims to support change management through a cross-domain model, in this case integrating information about requirements, functions, components, deliverables and detail design activities. Changes can be introduced to information in any of these interlinked domains and the ISF helps a designer to assess how the change might propagate within and across them.

Secondly, in terms of models aimed at generic knowledge description, some formalisms are mainly oriented around describing a process as a flow of activities while also indicating the product elements used as input and output to those activities, alongside selected links in the product domain. For example, the well-known IDEF3 Process Description Capture Method and formalism aims to support capture of different user perspectives on precedence and causality relationships in a process (Mayer *et al.* 1995). IDEF3 is based on two connected diagram types: the process flow diagram and the object state transition network diagram. The former represents a process through a flow of units of behaviour (UOBs, i.e., actions), which are interlinked via precedence links and junctions (i.e., logic operators such as And, Or and Xor). The latter diagram represents an object which is transformed from one object state to another through the UOBs. The Applied Signposting Model (ASM) developed by Wynn *et al.* (2006) is targeted specifically at modelling engineering design processes. It represents design as structured activity networks in which tasks cannot interact directly, but are interconnected via input and output deliverables. The deliverables refer to design parameters and their status at each point in the process, and parameters can in turn be interconnected into a product information hierarchy. The ASM was developed to support several defined purposes, and following case studies it was

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

found that the integrated nature helped to achieve a comprehensive representation by requiring users to consider the rationale behind task sequences in detail.

Thirdly, other formalisms aim to capture design and development process knowledge in a complete and contextualised way. As well as product and process elements, this involves representation of design rationale, design alternatives, and/or the history of a design as it evolves during the design process. Objectives for such modelling include capturing design knowledge to support its reuse (e.g., Abramovici & Chasiotis 2002); providing an enabler towards AI in design and intelligent CAD (e.g., Klein 2000); and coordinating work and decisions among design actors – for instance by analysing conflicts and propagating decision consequences (e.g., Arndt & Klein 2002) or by facilitating reactive control considering the changing design context (Robin *et al.* 2007). While some such work has an ultimate aim to support practice, other authors including Wang *et al.* (2013) and Grebici *et al.* (2009) concentrate on using integrated formalisms to model and analyse the relationship between product and design process, with the aim to derive research insights.

One representative example of a formalism in this category is FBS-PPRE (Bernard *et al.* 2006) which aims to capture and reuse the description of various information objects in PD, including process history and alternatives, with the intention to support managing their evolution and performance. The formalism extends the function, behaviour and structure paradigm to process, product and resource. The design process is treated as the transformation of functions to behaviours and finally to structure. Function is extended to process (temporal, spatial and hierarchical organisation of activities), product (object resulting from the process), resource (means used in a process), and external factors (contextual and environmental effects). Behaviour represents a set of laws governing the evolution or sequence of changes that takes place. Structure represents the decomposition of the system into its basic components. Pro2Kon (Abramovici & Chasiotis 2002; Chasiotis 2006) is another formalism that aims to represent multiple aspects of implicit and explicit procedural knowledge, with the objective to reuse that knowledge in support of distributed product development. Pro2Kon allows calculation methods, models, and results to be represented and interlinked by integrating four main model categories. The explicit procedural knowledge is represented using model elements concerning process knowledge and design-oriented product knowledge, while implicit knowledge is formalised in terms of elements that represent behaviour-oriented product knowledge and configuration knowledge. Similarly the *Co*llaborative*Mo*del (CoMoDe) formalism views design processes as a set of iterative activities from which initial design specifications evolve to a final desired product (Gonnet *et al.* 2007). Design activity is represented in terms of how the 'DesignObjects' evolve during the design process through a set of 'Operations' applied by 'Actors' given a set of 'Requirements', creating instances in time called 'ModelVersions'. 'Activities' can be further detailed to trace the historical, contextual and rationale relationships between different 'ModelVersions'.

To populate Table 4 considering these and other formalisms revealed by the review, purposes from Table 2 were initially assigned to each publication during a workshop with the majority of coauthors and later revalidated as described in Section 1.3, leading to improvements in the classification. The content of Table 4 distinguishes between primary and secondary purposes of each publication.

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

The primary purpose indicates that the work was specifically developed for this purpose, at least as emphasised in the corresponding publication. Where a secondary purpose is noted, this indicates that although the original authors did not emphasise that a formalism was developed for that purpose, the workshop participants and other authors believed it could arguably be used for it (or is used for it) in practice. A clear differentiation between primary and secondary purposes might not always be possible or at least may remain a subject for debate. However, we believe that this distinction is important as it indicates not only the original objective as emphasised by a paper's authors, but also potential capabilities of each approach. In particular, modelling formalisms are often taken up by other researchers or practitioners, who see the potential of the formalism for another purpose or only use a part of the originally included functionality. For example ASM was originally developed in the context of a planning support approach (Wynn *et al.* 2006), and has since been applied for process visualisation and improvement (e.g., Zhang, Hao & Thomson 2015).

Finally, although care was taken in formulating Tables 3 and 4, including several iterations among this paper's authors to refine the content, it is important to note that approaches (and especially their intended purposes) are sometimes described in an ambiguous or incomplete way. Thus subjectivity could not be eliminated entirely from Tables 3 and 4. It should further be noted that Table 4 is not intended to imply a ranking of formalisms according to the number of purposes covered, because purposes are addressed in different ways and, arguably, to different degrees of research validation.

5. Discussion

5.1. Potential advantages and challenges of more integrated modelling

Industry would arguably benefit from more interconnected model instances which would help to carry out engineering activities in an integrated fashion, executed and coordinated by pulling information more seamlessly from a heterogeneous set of models. Coupled and integrated model instances connect heterogeneous information in an explicit way, which could help to identify and consider the relevant information when making decisions. Model instances that link product and design process potentially make explicit how process activities depend on and lead to the generation of product properties and their descriptions. This could highlight misalignments between product and process structure and help companies to determine how activities can be bundled in a way that makes sense from a product perspective. Other applications of integrated modelling identified in the literature are summarised in the cells of Table 4. However, it should be noted that many of the publications we reviewed focus more on specifying a formalism than on describing the detail of its application.

One important consideration is whether the effort that would be required to construct integrated model instances can be justified. If integrated model instances were to be generated from scratch, the effort would clearly be substantial in many situations. Coupled model instances might in principle be easier to develop and maintain, because each model instance in a coupled set can be created and updated separately. On the other hand, an inherent limitation of a coupled approach to integrating model instances is that the links must typically be created

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

Table 4. Categorisation by purpose of selected integrated formalisms. Primary purposes are specifically emphasised in the indicated publication. Secondary purposes (i.e., possible applications of the formalism suggested by the authors of this article) are shown in italic. See Table 3 for a detailed description of each formalism, and Table 2 for a detailed explanation of each purpose

	Project visualisation	Project planning	Project execution and control	Project development	Product visualisation	Product synthesis	Product analysis and evaluation	Product life cycle support
Examples of integrated formalisms introduced in the context of approaches that exploit integrated product–process information								
Signposting, Clarkson & Hamilton (2000)		*Analyse process performance and improve planning accordingly*	Guide designer to next task				*Determine impact of product maturity on task sequence*	
DEPNET, e.g., Ouertani & Gzara (2008)		Plan design change process	*Identify activities which require rework after engineering changes*		Visualise data dependencies			
APDP, Lévárdy & Browning (2009)		Generate ideal task sequence; Provide decision support for budgeting and scheduling	Monitor progress through set of metrics				*Reduce technical performance risks*	

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

Table 4. (continued)

ISF, Ahmad *et al.* (2013)		Assess design change process	Capture process knowledge relevant to design change	Capture product knowledge relevant to design change
Examples of integrated formalisms introduced for general knowledge representation				
IDEF3, Mayer *et al.* (1995)	Capture user perspectives on precedence and causality relationships	*Support project planning based on documentation*	Describe hierarchical processes according to different user perspectives	*Product documentation through different states of development*
Design History System (Shah *et al.* 1996)			Capture and reuse design history	
SHARED (Gorti *et al.* 1998)			Capture comprehensive design knowledge	

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

Table 4. (continued)

Engineering History Base (Taura & Kubota 1999)				Store, refer to and reuse engineering from past cases				
Design Roadmap (Park & Cutkosky 1999)	Provide information display adjusted to task at hand		Coordinate collaboration, conflict management	Describe large-scale mature design processes; Select and design local process automation				
MOKA (Klein 2000; Arndt & Klein 2002)		Make recommendations on how to proceed	Support conflict management		Support interoperation; propagate consequences			
DMS (Whitfield *et al.* 2002)		*Support decoupling of design activities and resequencing*	Support process coordination of distributed design activities			*Track Pareto optimality of design to satisfy goals and requirements*	*Support design reuse for future variant designs*	

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

Table 4. (continued)

Topological structures (Braha & Reich 2003)			Describe design processes and improve understanding of phenomena like design failure and knowledge bottlenecks	
Connectivity Map (Yassine *et al.* 2003)	*Plan an efficient way to carry a process taking into account product architecture*	Evaluate and analyse the relationship between process and product		*Assess the ability of product's architecture to support changes*
GRAI (Merlo & Girard 2004)	Structure or reengineer the design process	Coordinate engineering, manage tasks allocation and scheduling	Specify information system to provide required information	

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

Table 4. (continued)

Baxter *et al.* (2007)			Knowledge structuring, retrieval and reuse		
IPPOP (Robin *et al.* 2007)	Plan and reactively control collaboration between actors	Allocate resources effectively	Capture knowledge for reuse		
OPM applied to SE (Sharon *et al.* 2013)	Automatically generates a plan taking into account project activities and timing with product's structure	Updates and replans the project if any changes are made on the product side		*Capture product elements' functional, structural and behavioural aspects*	Documentation of product in a hierarchical model
Pro2Kon, Abramovici & Chasiotis (2002), **Chasiotis** (2006)			Continuous improvement by documenting engineers suggestions	Capture and provide information about product structure and behaviour	Product documentation and knowledge management

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

Table 4. (continued)

FBS-PPRE, Bernard *et al.* (2006)			Automating the redesign process	Capture and reuse design process history and alternatives		Life cycle management by extending FBS to product, process and resource
ASM, Wynn *et al.* (2006)	Generate filtered views	Generate plans from process knowledge		Capture expert knowledge		
CoMoDe, Gonnet *et al.* (2007)			Conflict detection and resolution	Represent, capture and trace design process history for reuse and learning	Capture requirements and decision rationale	Knowledge management for reuse and learning purposes

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

Table 4. (continued)

Integrated diagrammatic model (Grebici *et al.* 2009)		Study information use in design processes		
DRed 2.0, Aurisicchio & Bracewell (2013)	Express the plan		Assist decision making	Record design rationale

manually. An automated or semi-automated approach would be desirable, and might be possible in some cases, for example where the various model instances have an underlying common ontology and hierarchical structure. Another issue to consider regarding integration through coupling of existing models is that much of the insight in modelling is gained through the process of building models, and in practice many models that are useful in their original context may not be considered complete by their creators, and may not be updated as the situation they represent continues to evolve. It may therefore be difficult to repurpose models at a later date (or to interpret them within a coupled system of models) if the original context is not appreciated.

As well as justifying the increased effort, another challenge in achieving greater integration is managing the increased volume of information that integrated product and process representation implies. Unless integrated or coupled model instances are expressed on a very high level of abstraction, they are likely to be larger and therefore more difficult to visualise and navigate than isolated models. For example, an integrated model based on graphical diagrams can be expected to involve more boxes and arrows than an equivalent diagrammatic model covering a single domain. In consequence the scope of modelling may need to be more constrained if an integrated model is to be visualised and manipulated in the same way as a single-domain model. A number of authors have accordingly noted that to apply integrated modelling on a large scale, advanced visualisation techniques are needed. Approaches that have been suggested to address this include thematic-specific views and structures (Munker *et al.* 2014); filtering of model data to generate partial views (e.g., Wynn *et al.* 2006; Lindemann *et al.* 2009); hierarchical structures allowing opening and closing of model elements (e.g., Wynn *et al.* 2006; Sharon *et al.* 2013); and spreading a model over many worksheets connected using bidirectional hyperlinks (Aurisicchio & Bracewell 2013). However, these approaches require specialised tools that may not always be available in practice.

5.2. Progress towards integrated modelling in research literature

Table 4 suggests that key approaches, frameworks and formalisms in the research literature to date do not thoroughly address the possibilities for greater integration across modelling domains and model instances. In particular, Table 4 indicates that none of the selected formalisms cover all purposes that research has considered, although each purpose is addressed by at least one publication. The largest number of entries in the product columns of Table 4 fall under *product life cycle support*. Many of the formalisms require information that would not normally be available as part of a product development process. For example, practitioners may not have ready definitions for the parameter confidence levels that would be required for the Signposting model, so they would need to generate them as part of thinking through the process if they wanted to apply this approach. Considering the approaches that propose integrated formalisms for general knowledge capture, the formalisms are typically very broad in scope yet where application studies are reported, they are relatively selective focusing on a small aspect of the overall problem space that the approaches are claimed to address.

Notably many of the approaches, frameworks and formalisms that were reviewed are not integrated with standard product representations such as

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

parametric solid models or Bills of Materials (BOM). This might be partly explained by the difficulty for researchers of gaining access to data and software tools required. Another issue here is that it may often be difficult to reconcile the hierarchical structures of the models that need to be integrated. For example, an assembly BOM groups components according to the assembly activities. Some systems arrive preassembled from the suppliers, while others are put together during the assembly process and therefore are entered explicitly in the assembly BOM, so that engines and rivets might appear at the same level of detail in the assembly BOM. However, the tasks associated with designing or validating these parts can have a quite different structure. While the rivets in this example are likely to be carry-over parts with no tasks associated with them, the engine might have tasks associated with its specification and validation, if provided by a supplier, or many more tasks, if it is designed in-house. Therefore, there would need to be a many-to-many relationship between tasks and components in this example to represent them within the same model instance. This could cause a potentially complex modelling task with a dense structure of interdependencies to be managed. One of the challenges is to find suitable breakdowns at different levels of detail which allow composition and decomposition of elements, because many subtasks or components could be part of several tasks or systems. In practice companies work around this kind of issue through practical modelling conventions, for example by assigning components to systems based on the team structure in which they are designed, and individuals keep track of additional links. An integrated approach would seem to require capturing those links more explicitly.

There are many activities that could be supported through integrated models that are not comprehensively addressed in current research literature. For example, as we argued in Section 1.1, designers frequently need to trade product cost or performance against time and cost of the design process. Possibilities may arise to reduce product cost, e.g., by optimising the design, but this task would also affect the overall process and its resource utilisation. In other words, it would be useful to assess how the design process affects product quality, to assess how much more effort might be required to design an improved product. Making such assessments requires good overview over both the product and its design process. This often depends on the tacit knowledge of individuals or on calculations for specific scenarios. It could arguably be aided by development or reuse of integrated models which could help to capture and navigate relevant knowledge and thereby help to avoid overlooking important issues, such as the impact a product decision has on the allocation of resources to other tasks that require attention from the same people.

5.3. Progress towards integrated modelling in industry

The uptake of integrating modelling approaches in industry appears limited to date. IDEF has been available since the 1980s and is used in industry; however, it is rarely implemented as completely as described in the method documentation. The research systems have been validated to various extents ranging from theoretical discussions, through simplified demonstrative case studies, to industrial deployments. However, the wider industrial uptake is limited. Some approaches that originated in research have been rolled out through industrial collaborations (e.g., Aurisicchio & Bracewell 2013) and/or in teaching

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

(e.g., Albers & Braun 2011). Certain commercial software tools also support integrated modelling. For example, widely used diagramming tools are flexible enough to allow representation of information across multiple domains, while some integrated formalisms such as IDEF and MDM are implemented in special-purpose commercial tools. There is arguably a need for more empirical work to investigate the issues surrounding integrated modelling in industry, to determine whether currently available software is adequate and what improvements might be needed.

Another consideration for further work is to investigate the extent to which model integration is desirable or appropriate. In principle it would be desirable to ensure that models are sufficiently interconnected or integrated that they might be used as a reference to avoid important oversights that fall between the competencies of the different people involved. However, as we have noted the effort for most companies to achieve such integration would likely be significant, and the benefits difficult to quantify. Integration of models in practice is likely to require the introduction of new modelling approaches or changes to existing ones, a transition that could imply significant cost and effort. Maintaining an integrated model might also require considerable additional effort when compared to a collection of isolated models, because changes to either the product or the process representation might also lead to changes in the other domain. This could potentially be advantageous as it would help to consider the consequences of changes in a holistic way. However, in an environment where time pressure is significant, for example during later stages of a project, it might be difficult to justify the effort to keep such models up to date. The pragmatic approach is to bring models and information together in a way that allows decision makers to gain a useful overview and reason about the connection between products and processes, while accepting that this may not cover all issues that can be envisaged and that complete consistency may be difficult to achieve.

6. Conclusions

While many modelling approaches exist for products and their design processes, to date relatively few address the integration of these two domains. This article has developed a terminology for describing the different forms and levels of integration that are possible when modelling products and their design processes in an integrated fashion. A review of selected approaches published in the academic literature summarises some of the main integrated models and highlights that none of them individually address all issues identified in this article.

To date, integrated modelling of products and processes seems to be a rather theoretical approach that draws attention to connections between product and process information that might be tacitly understood, but that are rarely shown explicitly at present. In terms of model instances, some advantages and disadvantages of coupling vs. integration as strategies to represent the links between these domains have been suggested. These issues could be further analysed in future, perhaps through empirical work, to investigate which approach might be preferable with respect to particular situations and applications. In terms of formalisms, most integrated formalisms we reviewed are research systems, which even where evaluated in industry do not have a broad uptake to date. Several commercial systems do offer the possibility to integrate information more usually

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

captured in separate model instances; however, we did not find empirically derived conclusions about the value of these approaches to support engineering practice.

To bring integrated product and design process modelling as considered in the research literature closer to the industry context, we suggest several challenges need to be overcome:

(i) The scope and focus of integrated models needs to be clearly defined and articulated; it is hoped the vocabulary introduced in this article may contribute to this discussion.
(ii) Practical guidelines are needed regarding how to develop and maintain integrated models, for instance how the issue of different hierarchies should be handled.
(iii) Visualisation techniques and tools need to be developed to help practitioners work with the larger volumes of information that would be required in more integrated models, and to help retain sufficient consistency.

This article is based on literature study, informed by the authors' experience and observations during previous case studies. As such our conclusions represent an integration of insights in the literature along with new insights developed by considering the issues other researchers have raised. In future we hope to undertake empirical work to further explore the potential benefits and limitations of integrated modelling for different situations. The creation of more integrated models could potentially require considerable additional effort, both to build and to maintain model instances. Nevertheless the authors believe the opportunities to benefit from integrated models are sufficient to warrant additional research.

Financial Support

This research received no specific grant from any funding agency, commercial or not-for-profit sectors.

References

Abramovici, M. & **Chasiotis, C.** 2002 Integrated documentation of procedural knowledge in product development. In *Proceedings of Integrated Product and Process Development (IPPD) 2002, Nov. 21–22, 2002, Wroclaw, Poland*.

Ahmad, N., **Wynn, D. C.** & **Clarkson, P. J.** 2010 When should design changes be allowed to accumulate? In *Proceedings of IDMME – Virtual Concept 2010, Bordeaux, France, October 20–22*. Springer.

Ahmad, N., **Wynn, D. C.** & **Clarkson, P. J.** 2013 Change impact on a product and its redesign process: a tool for knowledge capture and reuse. *Research in Engineering Design* **24** (3), 219–244.

Albers, A. & **Braun, A.** 2011 A generalized framework to compass and to support complex product engineering processes. *International Journal of Product Development* **15** (1–3), 6–25.

Ariyo, O. O., **Eckert, C. M.** & **Clarkson, P. J.** 2009 Challenges in identifying the knock-on effects of engineering change. *International Journal of Design Engineering* **2** (4), 414–431.

Arndt, H. & **Klein, R.** 2002 An ontology-based collaborative framework for decision support in engineering. In *Digital Enterprise Challenges: Life-Cycle Approach to Management and Production* **(ed. G. L. Kovács, P. Bertók & G. Haidegger)**, pp. 369–381. Springer.

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

Aurisicchio, M. & **Bracewell, R.** 2013 Capturing an integrated design information space with a diagram-based approach. *Journal of Engineering Design* **24** (6), 397–428.

Baxter, D., **Gao, J.**, **Case, K.**, **Harding, J.**, **Young, B.**, **Cochrane, S.** & **Dani, S.** 2007 An engineering design knowledge reuse methodology using process modelling. *Research in Engineering Design* **18** (1), 37–48.

Bernard, A., **Labrousse, M.** & **Perry, N.** 2006 LC universal model for the enterprise information system structure. In *Innovation in Life Cycle Engineering and Sustainable Development* **(ed. D. Brissaud, E. Tichkiewitch & P. Zwolinski)**, pp. 429–446. Springer.

Box, G. E. P. & **Draper, N. R.** 1987 *Empirical Model Building and Response Surfaces*. John Wiley & Sons.

Braha, D. & **Reich, Y.** 2003 Topological structures for modeling engineering design processes. *Research in Engineering Design* **14** (4), 185–199.

Browning, T. R. 2001 Applying the design structure matrix to system decomposition and integration problems: a review and new directions. *IEEE Transactions on Engineering Management* **48** (3), 292–306.

Browning, T. R. 2016 Design structure matrix extensions and innovations: a survey and new opportunities. *IEEE Transactions on Engineering Management* **63** (1), 27–52.

Browning, T. R. & **Eppinger, S. D.** 2002 Modeling impacts of process architecture on cost and schedule risk in product development. *IEEE Transactions on Engineering Management* **49** (4), 428–442.

Browning, T. R. & **Ramasesh, R. V.** 2007 A survey of activity network-based process models for managing product development projects. *Production and Operations Management* **16** (2), 217–240.

Bucciarelli, L. L. 1994 *Designing Engineers*. MIT Press.

Cartwright, N. 2003 Two theorems on invariance and causality. *Philosophy of Science* **70** (1), 203–224.

Chasiotis, C. 2006 Prozessbegleitende Wissensdokumentation und integrierte Wissensvisualisierung in der Digitalen Produktentwicklung. PhD dissertation, Shaker Verlag University Bochum, Aachen, Germany (in German).

Chua, D. K. H. & **Hossain, M. A.** 2012 Predicting change propagation and impact on design schedule due to external changes. *IEEE Transactions on Engineering Management* **59** (3), 483–493.

Clarkson, P. J. & **Hamilton, J. R.** 2000 'Signposting', a parameter-driven task-based model of the design process. *Research in Engineering Design* **12** (1), 18–38.

Danilovic, M. & **Browning, T. R.** 2007 Managing complex product development projects with design structure matrices and domain mapping matrices. *International Journal of Project Management* **25** (3), 300–314.

Demoly, F., **Yan, X. T.**, **Eynard, B.**, **Gomes, S.** & **Kiritsis, D.** 2012 Integrated product relationships management: a model to enable concurrent product design and assembly sequence planning. *Journal of Engineering Design* **23** (7), 544–561.

DoDAF Architecture Framework Version 2.02 (2010) US Department of Defense, http://dodcio.defense.gov/Library/DoDArchitectureFramework.aspx.

Eckert, C. M. & **Clarkson, P. J.** 2010 Planning development processes for complex products. *Research in Engineering Design* **21** (3), 153–171.

Eisenbart, B., **Gericke, K.** & **Blessing, L.** 2011 A framework for comparing design modelling approaches across disciplines. In *Proceedings of 18th International Conference on Engineering Design*, pp. 344–355. Design Society.

Eppinger, S. D., **Whitney, D. E.**, **Smith, R. P.** & **Gebala, D. A.** 1994 A model-based method for organizing tasks in product development. *Research in Engineering Design* **6** (1), 1–13.

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

Flanagan, T. 2006 Supporting design planning through process model simulation. PhD thesis, Cambridge University Engineering Department.

Frigg, R. 2003 Re-representing scientific representation. PhD thesis, Department of Philosophy Logic and Scientific Method, London School of Economics.

Gärtner, T., **Rohleder, N.** & **Schlick, C. M.** 2009 DeSiM: a simulation tool for project and change management on the basis of design structure matrices. In *Proceedings of the 11th International DSM Conference, Greenville, South Carolina, USA*. Carl-Hanser.

Gericke, K. & **Blessing, L.** 2012 An analysis of design process models across disciplines. In *Proceedings of DESIGN 2012, the 12th International Design Conference, Dubrovnik, Croatia, Vol. 1*, pp. 171–180. Design Society.

Gericke, K., **Eckert, C. M.** & **Wynn, D. C.** 2016 Towards a framework of choices made during the lifecycles of process models. In *Proceedings of DESIGN 2016, the 14th International Design Conference, Dubrovnik, Croatia*, pp. 1275–1284. Design Society.

Giere, R. N. 2004 How models are used to represent reality. *Philosophy of Science* **71** (5), 742–752.

Gonnet, S., **Henning, G.** & **Leone, H.** 2007 A model for capturing and representing the engineering design process. *Expert Systems with Applications* **33** (4), 881–902.

Gorti, S. R., **Gupta, A.**, **Kim, G. J.**, **Sriram, R. D.** & **Wong, A.** 1998 An object-oriented representation for product and design processes. *Computer-aided Design* **30** (7), 489–501.

Grebici, K., **Wynn, D. C.** & **Clarkson, P. J.** 2009 Describing information use in engineering design processes using a diagrammatic model. In *Proceedings of ICED 09, the 17th International Conference on Engineering Design, Palo Alto, CA, USA, 24–27 August 2009, Vol. 1*, pp. 571–582. Design Society.

Heisig, P., **Caldwell, N. H. M.** & **Clarkson, P. J.** 2014 Core information categories for engineering design – contrasting empirical studies with a review of integrated models. *Journal of Engineering Design* **25** (1–3), 88–124.

Heisig, P., **Clarkson, P. J.**, **Hemphälä, J.**, **Wadell, C.**, **Norell-Bergendahl, M.**, **Roelofsen, J.**, **Kreimeyer, M.** & **Lindemann, U.** 2009 Challenges and Future Fields of Research for Modelling and Management of Engineering Processes. Cambridge University Engineering Department, Technical Report CUED/C-EDC/TR. 148.

Klein, R. 2000 Knowledge modeling in design—the MOKA framework. In *Artificial Intelligence in Design '00* **(ed. J. S. Gero)**, pp. 77–102. Springer.

Kreimeyer, M. & **Lindemann, U.** 2011 *Complexity Metrics in Engineering Design: Managing the Structure of Design Processes*. Springer.

Kurakawa, K. 2004 A scenario-driven conceptual design information model and its formation. *Research in Engineering Design* **15** (2), 122–137.

Lévárdy, V. & **Browning, T. R.** 2009 An adaptive process model to support product development project management. *IEEE Transactions on Engineering Management* **56** (4), 600–620.

Lindemann, U., **Maurer, M.** & **Braun, T.** 2009 *Structural Complexity Management: An Approach for the Field of Product Design*. Springer.

Maier, J. F., **Wynn, D. C.**, **Biedermann, W.**, **Lindemann, U.** & **Clarkson, P. J.** 2014 Simulating progressive iteration, rework and change propagation to prioritise design tasks. *Research in Engineering Design* **25** (4), 283–307.

Mayer, R. J., **Menzel, C. P.**, **Painter, M. K.**, **Dewitte, P. S.**, **Blinn, T.** & **Perakath, B.** 1995 Information integration for concurrent engineering (IICE) IDEF3 process description capture method report, Knowledge Based Systems Inc. Technical Report KBSI-IICE-90-STR-01-0592-02, College Station, TX.

Melo, A. F. 2002 A state-action model for design process planning. PhD thesis, Department of Engineering, University of Cambridge.

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

Merlo, C. & **Girard, Ph.** 2004 Information system modelling for engineering design co-ordination. *Computers in Industry* **55** (3), 317–334.

Munker, F., **Albers, A.**, **Wagner, D.** & **Behrendt, M.** 2014 Multi-view modeling in SysML: thematic structuring for multiple thematic views. In *Procedia Computer Science, Vol. 28*, pp. 531–538.

O'Donovan, B., **Eckert, C.** & **Clarkson, P. J.** 2004 Simulating design processes to assist design process planning. In *Proceedings of ASME 2004 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Volume 3a: 16th International Conference on Design Theory and Methodology, Salt Lake City, Utah, USA, September 28–October 2, 2004*. American Society of Mechanical Engineers (ASME).

Ouertani, M. Z. & **Gzara, L.** 2008 Tracking product specification dependencies in collaborative design for conflict management. *Computer-Aided Design* **40** (7), 828–837.

Park, H. & **Cutkosky, M. R.** 1999 Framework for modeling dependencies in collaborative engineering processes. *Research in Engineering Design* **11** (2), 84–102.

Potter, S., **Culley, S. J.**, **Darlington, M. J.** & **Chawdhry, P. K.** 2003 Automatic conceptual design using experience-derived heuristics. *Research in Engineering Design* **14** (3), 131–144.

Reich, Y. & **Subrahmanian, E.** 2015 Designing PSI: an introduction to the PSI framework. In *DS 80-2 Proceedings of the 20th International Conference on Engineering Design (ICED 15), Vol. 2: Design Theory and Research Methodology: Design Processes, Milan, Italy, July 27–30, 2015*, pp. 137–148. Design Society.

Robin, V., **Rose, B.** & **Girard, Ph.** 2007 Modelling collaborative knowledge to support engineering design project manager. *Computers in Industry* **58** (2), 188–198.

Shah, J. J., **Jeon, D. K.**, **Urban, S. D.**, **Bliznakov, P.** & **Rogers, M.** 1996 Database infrastructure for supporting engineering design histories. *Computer-Aided Design* **28** (5), 347–360.

Sharon, A., **de Weck, O. L.** & **Dori, D.** 2013 Improving project–product lifecycle management with model–based design structure matrix: a joint project management and systems engineering approach. *Systems Engineering* **16** (4), 413–426.

Smith, R. P. & **Morrow, J. A.** 1999 Product development process modelling. *Design Studies* **20** (3), 237–261.

Stachowiak, H. 1973 *Allgemeine Modelltheorie*. Springer.

Star, S. L. 1989 The structure of ill-structured solutions: boundary objects and heterogeneous distributed problem solving. In *Distributed Artificial Intelligence, Vol. 2* **(ed. M. Huhns & L. Gasser)**. Morgan Kaufman, Menlo Park, CA.

Suppe, F. 1977 The search for philosophic understanding of scientific theories. 2nd edn **(ed. F. Suppe)**. The Structure of Scientific Theories. University of Illinois Press.

Taura, T. & **Kubota, A.** 1999 A study on engineering history base. *Research in Engineering Design* **11** (1), 45–54.

Wang, W., **Duffy, A.**, **Boyle, I.** & **Whitfield, R.** 2013 Creation dependencies of evolutionary artefact and design process knowledge. *Journal of Engineering Design* **24** (9), 681–710.

Whitfield, R. I., **Duffy, A. H. B.**, **Coates, G.** & **Hills, W.** 2002 Distributed design coordination. *Research in Engineering Design* **13** (4), 243–252.

Wynn, D. C. & **Clarkson, P. J.** 2005 Models of designing. In *Design Process Improvement: A Review of Current Practice* **(ed. P. J. Clarkson & C. M. Eckert)**, pp. 34–59. Springer.

Wynn, D. C., **Eckert, C. M.** & **Clarkson, P. J.** 2006 Applied signposting: a modeling framework to support design process improvement. In *Proceedings of ASME 2006 International Design Engineering Technical Conferences and Computers and*

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press

Information in Engineering Conference, Vol. 4a: 18th International Conference on Design Theory and Methodology, Philadelphia, Pennsylvania, USA, September 10–13, pp. 553–562. American Society of Mechanical Engineers (ASME).

Wynn, D. C. & **Eckert, C. M.** 2017 Perspectives on iteration in design and development. *Research in Engineering Design* **28** (2), 153–184.

Yassine, A., **Whitney, D.**, **Daleiden, S.** & **Lavine, J.** 2003 Connectivity maps: modeling and analysing relationships in product development processes. *Journal of Engineering Design* **14** (3), 377–394.

Zhang, X., **Hao, Y.** & **Thomson, V.** 2015 Taking ideas from paper to practice: a case study of improving design processes through detailed modeling and systematic analysis. *IFAC-PapersOnLine* **48** (3), 1043–1048.

https://doi.org/10.1017/dsj.2017.2 Published online by Cambridge University Press