This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

A Design Framework for Social Product Development

Hannah Forbes, Dirk Schaefer, Jitesh Panchal, and Ji Han

***Abstract*—Social product development (SPD) is defined as a group of "coalescing tools and socio-technologies" represented by several tenants including crowdsourcing, internet-based mass collaboration, open innovation, and cloud-based design and manufacture. Existing examples of SPD have resulted in enhanced collaboration in design teams, shorter lead times, and significant reductions in R&D costs. Despite these potential benefits, guidance on how to conduct SPD is lacking in existing literature and the use of SPD tenants in the industry is limited. The aim of this article is to present a framework to guide the application of SPD. As part of this framework, the authors present a tool for the selection of SPD tenants that allows users to select an SPD tenant based on their requirements and capabilities. The framework allows practitioners to consider how SPD tenants can be used to solve design problems and allows the user to design SPD tenants with consideration of their own needs. This article concludes with a validation activity and discussion on future research directions in this field.**

***Index Terms*—Crowdsourcing, open innovation, social product development.**

I. Introduction

IN THIS section, the term social product development (SPD) is defined, and a case is made for its need and application in the realm of modern product development.

A. Defining SPD

In the existing literature, SPD is defined as a group of "coalescing tools and socio-technologies" represented by several tenants including crowdsourcing, Internet-based mass collaboration (IBMC), open innovation (OI), and cloud-based design and manufacture (CBDM) [1]. The definition of SPD is still evolving in literature and the authors judge the current definition to be too ambiguous in describing the term. Defining the complex and varied relationships that represent SPD, however, is difficult. For example, two people working on a product and communicating via Facebook messenger is not SPD but five people in a group chat on Facebook conducting an ideation activity is. The difference between SPD and product development, however, is not defined purely by the number of those involved, as one submission as part of a crowdsourcing process is still defined as SPD. It was this thought process and amalgamation of case studies that led the authors to define SPD by the only characteristic that all tenants of SPD share, which is the use of Web 2.0 technologies. Web 2.0 is defined as the "writable" phase of the World Wide Web [2] in which the responsibility of content creation is decentralized and all users have the opportunity to contribute and interact [3]. A product development process, therefore, becomes an SPD process when Web 2.0 technologies including social networking, instant messaging, information sharing, and other online communication such as voice over internet protocol (VoIP) are employed [4].

B. Need for SPD

The need for SPD can be expressed with three key points. First, teams in the current business environment are increasingly dispersed with technologies such as video conferencing making forming virtual teams a possibility. Virtual teams, however, make managing projects more complex [5]. SPD seeks to facilitate the project management process by "enhancing collaboration and communication" [6]. As Wu *et al.* [6] state "most successful product development teams have high levels of communication and collaboration" and Chui and Bughin [7] state that a "well-connected design network plays a vital role […] in design phases." The introduction of SPD tenants can, therefore, address this key concern in the current business climate.

Second, external input has been proven to benefit design teams and enhance the design problem-solving process. Abhari *et al.* [8] suggest that essential resources for innovation lie beyond the boundary of an organization and Bertoni *et al.* [9] state that "the development of technologically complex products requires a wide range of skills […] which are difficult to find within a single organization." As well as a need for external involvement, there is also a proven benefit to looking "beyond the walls" of a single organization. Thames and Schaefer [5] state that "innovation projects that are largely based on external development have shortened development times and need less investment." To support this statement, Huston and Sakkab [10} state that since the introduction of an SPD tenant at Procter and Gamble "productivity has increased by almost 60%."

Finally, in the current business climate, competitive advantage by incremental improvement alone is no longer possible [11].

Manuscript received March 29, 2019; revised August 4, 2019; accepted October 13, 2019. Review of this manuscript was arranged by Department Editor R. D. Evans. *(Corresponding author: Hannah Forbes.)*

H. Forbes and J. Han are with the Department of Industrial Design, University of Liverpool Faculty of Engineering, Liverpool L69 3GL, U.K. (e-mail: hannah.forbes@liverpool.ac.uk; ji.han@liverpool.ac.uk).

D. Schaefer is with the University of Liverpool, Liverpool L69 3GL, U.K. (e-mail: schaefdk@liverpool.ac.uk).

J. Panchal is with Purdue University, West Lafayette, IN 47907 USA (e-mail: panchal@purdue.edu).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEM.2019.2948996

Organizations must look for new ways to innovate to increase market share and satisfy "increasingly sophisticated customer needs" [12]. SPD is a "fundamentally new approach to innovation" [10] that offers a route to competitive advantage for organizations. Procter and Gamble's SPD initiative, connect and develop, resulted in "billions of dollars in revenue" and "35% of the company's innovations" are credited to the initiative. SPD is, therefore, a new approach to thriving in the current business climate.

II. Literature Review

The aim of this literature review is to identify SPD frameworks and provide evidence for research gaps. The term "framework" is used to describe an instructional set of principles or steps that guide the inclusion of SPD tenants in the product development process. Existing literature could, therefore, include an SPD framework in its entirety or a framework to guide the inclusion of a single tenant. The term SPD is sometimes used to describe the development of social products that generate social impact. Any literature relevant to this understanding of SPD is not included in this body of work.

To search for relevant literature, terms such as "SPD framework" and "SPD design framework" were used. In this relatively young field, however, the number of results were expectedly low. "OI framework" was, therefore, used as a search term to widen the search field. This term yielded literature presenting frameworks for individual tenants as well as frameworks for the application of SPD as a whole. The authors, therefore, recognized that this fragmentation of SPD and the formulation of frameworks on a lower level of abstraction is more common. As a consequence, this literature review includes first a consolidation of overall SPD frameworks and then looks at the literature that discusses frameworks for individual SPD tenants. It concludes with identification of the gaps within these individual sectors, as well as literature gaps at a higher level of abstraction.

The search term and variances of the term "SPD framework" yielded only one result. In this article, Forbes and Schaefer [13] introduce the idea of an SPD framework which presents the potential impact each SPD tenant may have on the product development process. It is designed, however, to be a precursor to a more refined SPD framework and does not provide any detailed instruction to the inclusion of SPD in the product development process.

Existing literature, for individual tenant frameworks, includes the "distributed team innovation" by Larsson *et al.* [14] for the use of IBMC in product development. This framework is for ensuring product development; education and research are given equal precedence in the product development process [14]. Bartl *et al.* [15] present a similar framework for the "co-creation" of products. They suggest that three aspects of the product development process require equal consideration: methods and tools, organization, and culture. While both of these frameworks offer insight into the management of the entire product development process, they are not instructional, nor detailed enough to effectively guide the adoption of SPD.

Frameworks for the application of OI in product development are also at a high-level of abstraction and do not provide a prescriptive and detailed approach to application. Peterson and Schaefer [1], however, suggest that an "OI culture" is required to foster SPD. This suggests that OI frameworks could act as a precursor to an SPD framework and ensure that an industrial environment is prepared to conduct an SPD approach. This idea, however, falls outside the scope of this article and is not developed further.

The existing literature on crowdsourcing frameworks for product development is relatively extensive and provides several insights into both high level and detailed requirements for an SPD framework. Niu *et al.* [16] present a framework for the application of crowdsourcing in product development, guiding the user through important crowdsourcing decisions. Panchal [17] also presents a framework for the use of crowdsourcing in product development, providing a four-step approach to crowdsourcing application. This framework includes three key steps: selecting crowdsourcing initiatives, making design decision, and incentive design. Panchal also provides further detail regarding "incentive design" by presenting a game-theoretic model for managing crowd participation. Similarly, Abrahmason *et al.* [18] present an "incentives mix framework" for understanding crowd participation and Cullina *et al.* [19] and Gerth *et al.* [20] provide in-depth research on finding the "qualified crowd" in crowdsourcing contests. Finally, Kittur *et al.* [21] consider the crowdsourcing of human intelligence tasks (HITs) and "provide a systematic and dynamic way to break down tasks into subtasks and manage the flow and dependencies between them."

While a relatively rich field of research, these crowdsourcing frameworks predominantly refer to "low-level" aspects of crowdsourcing application. For example, Cullina *et al.* [19] discuss the need to understand crowd motivation in contests which is a single factor contributing to the successful implementation of crowdsourcing. Crowdsourcing, however, is a single tenant of SPD and further high-level research is required to understand the place of crowdsourcing in this overall framework. Fig. 1 illustrates this.

With reference to this hierarchy, there are several research gaps, at different levels of abstraction.

1) An SPD framework is yet to be presented in the existing literature.
2) An opinion on SPD from practitioners is yet to be presented.

In addition to literature gaps in higher-level research, there are several literature gaps within research at a lower level of abstraction.

1) To date, the application of crowdsourcing is only considered in concept generation and evaluation.
2) Crowdsourcing contests are studied in greater detail than other forms of crowdsourcing such as open calls and HITs
3) Several factors for successful crowdsourcing are not considered such as how to frame design problems to the crowd.
4) There is a lack of validation across this literature field with the majority of research being at a conceptual stage.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

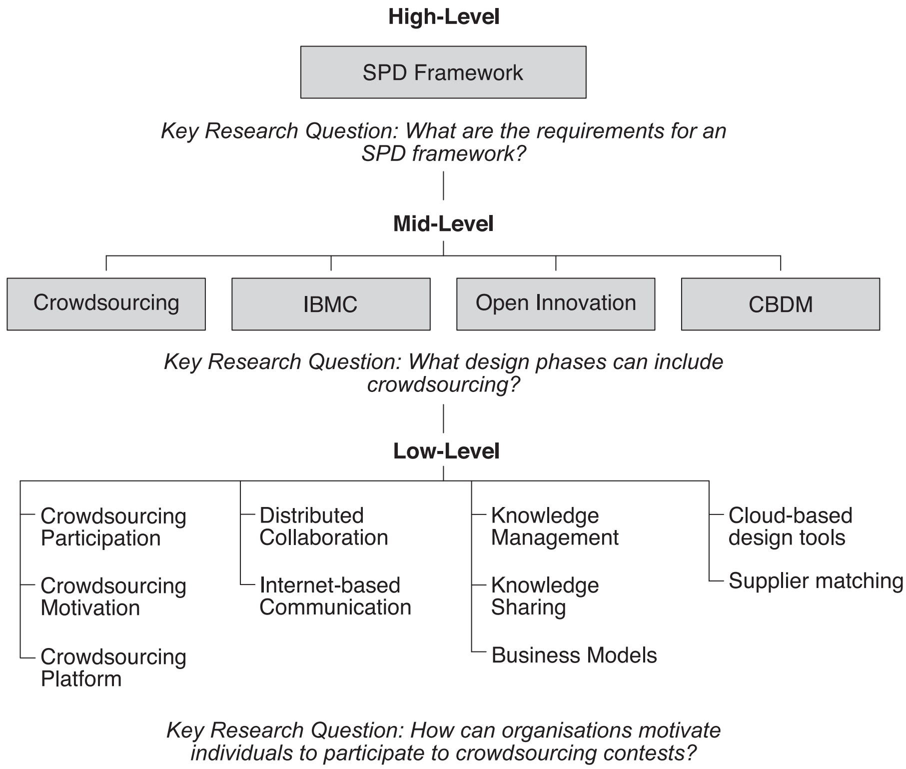

Fig. 1. Hierarchy of research in the context of an SPD framework.

The aim of this article is to address these research gaps by developing and presenting an SPD framework. With respect to this aim, the objectives are as follows:

1) identify how previous researchers have "designed design methods;"
2) select a design method to adapt;
3) adapt the design method to create an SPD framework.

This framework will allow users to identify an SPD tenant suitable for solving their design problem and support the design of this SPD tenant.

III. Methodology: Designing a Design Framework

To construct the methodology for conceiving an SPD framework, the authors sought literature for "designing a design framework." It was found that the conception process of design frameworks was rarely documented. For example, Panchal's [17] framework for the design of crowdsourcing contests presents a framework as a "step toward addressing this research gap" and the conception process is not described. Larsson *et al.* [14] present a framework for developing products with distributed teams that was conceived using an ethnographical study of practitioners. While the results of the ethnographical study are presented and analyzed in this literature, the specific process of using the experimental findings to conceive the presented framework is not described [14].

In an attempt to find a described and validated framework conception process, the authors looked beyond the field of design, using search terms such as "conceiving a framework" and "designing a framework". Mior *et al.* [30] discuss the process taken to conceive a framework for the "delivery of collaborative musculoskeletal care." The process taken was user-centric, with several rounds of requirements analysis including different stakeholders. Boydell *et al.* [31] also followed a similar approach described as a "multistage user consultation" for the conception of an "evaluation framework" for pediatric telepsychiatry. In both processes, the outcome of this in-depth requirements analysis was key themes that were carried into the conception of a design framework. In both circumstances, however, how the "key themes" were processed to form a framework was not described. Finally, Reeves *et al.* [34] design a framework for the professional development of school leaders and managers by analyzing and synthesizing literature in their field.

Existing literature, therefore, does not provide a prescriptive approach for the complete construction of a design framework. In the absence of a methodical approach to the conception of a design framework, the authors choose to adapt an existing design approach. In the following sections, existing design methods are compared and selected for use in this context. The selected method(s) are then adapted to produce a high-level SPD framework.

IV. Design Method Selection

In this section, established design methods are examined, selected, and compared for use as an SPD framework.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Fig. 2. Map of design method literature by Wynn and Clarkson [36].

A. Narrowing the Design Method Pool to Procedural Methods

Wynn and Clarkson [35], [36] present extensive and comprehensive literature on design methods. Their organizational framework, shown in Fig. 2, categorizes all development and design methods according to the method's scope and the type of method.

In order to reduce the design method pool, the authors began by considering which of the categories presented by Wynn and Clarkson [35], [36] would be appropriate for an SPD framework.

It was first determined that a procedural design method would be required. This is due to statements by Wynn and Clarkson and other academics on its suitability, above other types, for actual implementation. Wynn and Clarkson [36] state procedural methods "convey best practices intended to guide real-world situations" and Harmsen and Saeki [38] state that procedural methods are "most relevant to practical situations." Furthermore, analytical methods are described as "better positioned to provide support in specific contexts" [35] while abstract methods are "not able to provide insights for implementation" [36]. SPD tenants are used to solve a breadth of design problems which means methods designed for specific circumstances are not appropriate.

Having now refined the design method pool to procedural methods, the authors looked further into the distinction between micro-, meso-, and macroprocedural methods and which was most appropriate for an SPD framework. As stated by Wynn and Clarkson [36], microlevel, mesolevel, and macrolevel, in this context, are defined as the following.

1) Microlevel: Focuses on individual process steps and their immediate contexts.
2) Mesolevel: Focuses on the end-to-end flow of tasks as the design progresses.
3) Macrolevel: Focuses on project structures and/or the design process in the context.

TABLE I
ELEMENTARY TASKS FOR AN SPD FRAMEWORK

Elementary Task	Description
Problem Clarification	Users are required to clarify and de-contextualise the design task they are interested in solving with an SPD tenant.
Requirements Analysis	Users are required to identify why existing practises aren't working, and why.
Concept Generation	Users should have an understanding of which SPD tenants are available to them.
Concept Evaluation	Users then need to evaluate the available SPD tenants and select one to solve their design problem.

Microlevel procedural methods offer a step-by-step process that can be easily applied to a variety of contexts. They do not, however, provide adequate guidance for the full design process and are instead useful for application in individual design phases or tasks. Macrolevel procedural methods offer high-level guidance for the design process and beyond, providing step-by-step guidance for organizational and managerial tasks as well as design tasks. As a consequence, however, neither provides adequate nor detailed information on how to conduct the design phases. Mesolevel procedural design methods, however, offer guidance throughout the whole design process and provide information and guidance for each design phase. They are, therefore, the most suited for this context, providing both breadth and depth in design process support.

B. Selecting a Mesolevel Procedural Method

With reference to Fig. 2, there are six mesolevel procedural design methods. These are Evans' Design Spiral [40], Pahl and Beitz' Stage Model [41], French's Stage Model [42], VDI2221 Stage Model [43], Hubka's Stage model [44], and Ullman's Stage Model [45]. To select between them, the authors consulted design method selection literature. Both López-Mesa and Thomspon [46] and Braun and Lindemann [47] suggest comparing the elementary tasks of the approach with the elementary tasks of the design task to determine compatibility. In order to conduct this comparison, the authors first identified tasks vital for the application of SPD. These are presented in Table I.

Following concept evaluation, the user will need to conduct a detailed design phase for the selected SPD tenant. The specific steps required as part of this process, however, are likely to be dictated by the chosen SPD tenant. They are, therefore, not included in Table I.

Studying each of the proposed design methods alongside Table I provides further insight into the compatibility of each design method. The design methods are distinguished in three key ways:

1) by the way in which they guide the design realization process;
2) by their level of detail;
3) by their content.

First, Evan's Design Spiral [40] is not appropriate for this design task due to the process of design realization. Unlike the other five design methods, Evan's Design Spiral requires an initial identification of "the overall arrangement" [40] before developing the design through the realization of relationships between the constituent components of the design. This process is not appropriate for the selection of design of SPD tenants because very little is known about the overall format or structure of the SPD tenant prior to use. The design realization process of Evan's Design Spiral in comparison to the other five methods is illustrated in Fig. 3.

French's stage model [42] was deemed inappropriate for this design task because of its lack of detail relative to the other design methods. Furthermore, some important elementary tasks, as shown in Table I, such as requirements analysis and concept generation are not included in French's model. Hubka's model is also deemed inappropriate for this reason. The first step in Hubka's model is "given design constraints" [44], which means important steps including problem clarification and requirements analysis are omitted.

The design method pool was, therefore, refined to three methods: Pahl and Beitz [39], VDI 2221 [43], and Ullman's Stage Model [45]. While each of these methods provides slightly different content and is presented in slightly different formats, it was deduced that, especially in the case that the final method would be adapted, choosing between them was not necessary. The next section, therefore, demonstrates how a combination of these methods was created and selected for use.

V. SPD FRAMEWORK

An adaptation of these three mesoprocedural design methods is used to create the SPD framework. The elemental tasks, as described in Table I, are incorporated and represented as task clarification, requirements specification, and tenant selection and evaluation. For SPD tenant application, the early design phases are focused on the selection of an SPD tenant. Detailed form design then prompts decision-making on the content and structure of the selected SPD tenant. The final design phases prompt the launch of the SPD tenant and a results processing phase to determine whether the SPD tenant produced an appropriate solution to the design problem. The SPD framework is shown in Fig. 4. In the following sections, the tool used for the tenant selection and evaluation phase is presented.

A. Tenant Selection and Evaluation

A key task in the early design phases of the application of SPD is the selection of an SPD tenant. SPD is represented by a group of tenants, as defined in Table II, which are united by their use of Web 2.0 technologies. As individual tenants, however, they are applied to product development in different ways. For example, OI is defined as "the use of purposive inflows and outflows of knowledge to accelerate internal innovation" where "external knowledge plays an equal role to [..] to internal knowledge" [22]. IBMC, however, is defined as the collective action of members of a large group [23]. Actors in IBMC, therefore, could be within

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

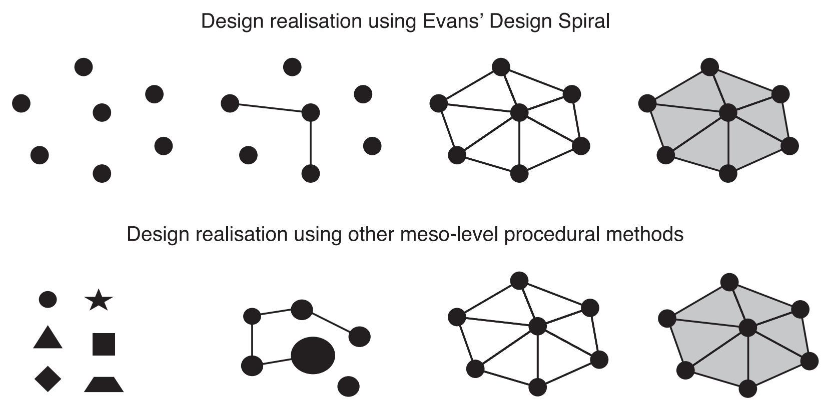

Fig. 3. Design realization in mesolevel procedural methods.

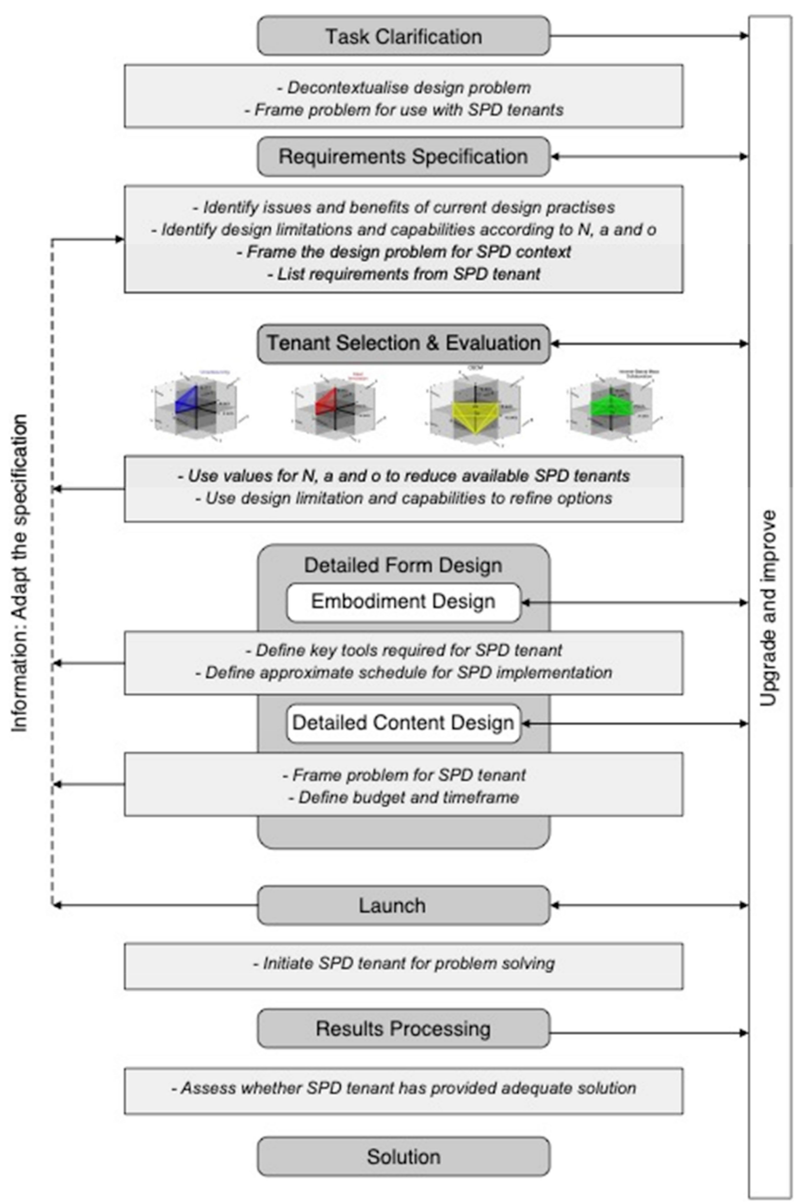

Fig. 4. Adapted design method for SPD tenant application using Pahl and Beitz [39], VDI2221 [43], and Ullman's Stage Model [45].

TABLE II
TENANTS OF SPD

SPD Tenant	Definition	Example
Crowdsourcing	Crowdsourcing is defined as the "the act of taking a job, traditionally performed by a designated agent […] and outsourcing it to a […] large group of people" [24] [17]. It is most regularly used in concept generation and concept evaluation of the product development process, and social networking is used to reach the intended crowd [25].	Organisation publishes and advertises a new product brief online and requests submissions
Internet-Based Mass Collaboration	Mass collaboration is a form of "collective action" where a large number of parties work together on a project [26] [27]. Each party makes a significant contribution to the project and all contributors are regarded as important [27]. Internet-based communication tools such as VoIP, file-sharing software and instant messaging are employed, and the number of actors must be over 25 [26].	Thousands of individuals around the world contribute expert knowledge to create an online encyclopaedia
Open Innovation	Open Innovation is defined as using knowledge inflows and outflows to fuel innovation [22]. Web 2.0 technologies such as social networks act as channels between internal teams and external knowledge sources.	Technology organisation publishes their code for a new app in an open-source community
Cloud-Based Design and Manufacture	Cloud-Based Design and Manufacture "is a type of parallel and distributed system consisting of a collection of inter-connected physical and virtualized service pools of design and manufacturing resources (e.g., parts, assemblies, CAD/CAM tools) as well as intelligent search capabilities for design and manufacturing solutions." [28]. A plethora of Internet-based communication tools are used including cloud-based software and CBDM process can include any number of actors.	CAD model is stored in cloud-based CAD software. Two mechanical engineers edit different parts of the model online and simultaneously

the same organization while OI specifies the involvement of knowledge external to the organization.

1) Selecting SPD Tenants According to Three Common Variables: The differences between the SPD tenants can be represented by three core variables, outlined in Table III. Proximity is defined as "nearness in space, time, and/or relationship" [29].

TABLE III
SPD VARIABLES AND THEIR METRICS

SPD Tenant Variable	Scale	Description
Organisational Proximity: Proximity to the organisation leading the product development project	1 - 10	1 = the actor or involved party is an external organisation that has minimal knowledge beyond what is in the public domain 5 = the actor or involved party is within the same industry as the organisation 10 = the actor or involved party is within the same organisation but not within the same location as the project lead (Web 2.0 technologies are therefore required for involvement)
Actor Proximity: Proximity to the other actors in the SPD tenant	1 - 10	1 = the actors or involved parties do not have a relationship and do not collaborate with each other 10 = the actors or involved parties are within the same organisations each other and have at least professional relationships with each other
Number of actors involved in the SPD tenant	1 - ∞	This refers to the number of people involved in the activity or the number of people with the opportunity to be involved in the activity

TABLE IV
SPD TENANTS AND THEIR VALUES FOR THE SPD VARIABLES

SPD Tenant	Proximity to organisation	Proximity to other actors	Number of actors
Crowdsourcing	1 - 5	1 - 5	25 - ∞
Internet-Based Mass Collaboration	1 - 10	1 - 10	25 – 100
Open Innovation	1 – 5	1 – 5	25 - ∞
Cloud-Based Design and Manufacture	1 – 10	1 - 10	1 - ∞

The main SPD tenants are assigned a variable range in Table IV. These numbers have been derived from the definitions of the SPD tenants given above.

To illustrate this, the scales given in Table IV are represented as three axes. The number of actors is assigned the γ axis, organizational proximity is assigned the χ axis, and actor proximity is assigned the Z axis. Placing the assigned values shown in Tables III and IV on these axes yields Figs. 5–8.

2) Quadrants of SPD: Figs. 5–8 present the eight quadrants of SPD. What these quadrants mean, beyond the context of the graphical representation, is described in Tables V and VI.

3) Using the Tenant Selection Tool as Part of the SPD Framework: In the task clarification and requirements specification

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

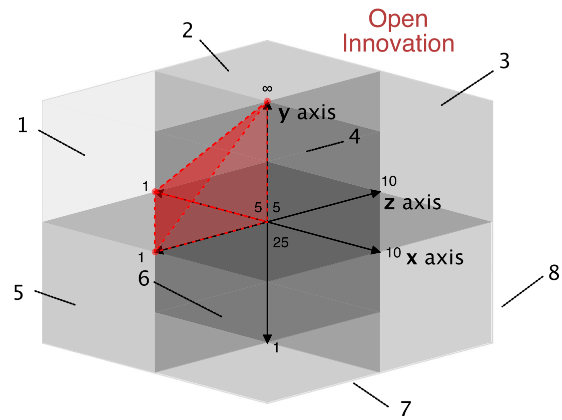

Fig. 5. Illustration of OI in the context of the three SPD variables.

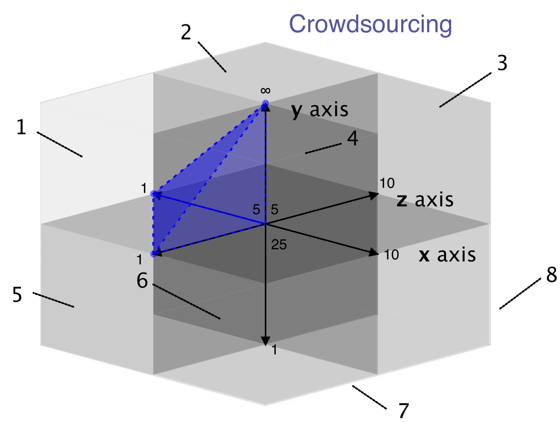

Fig. 6. Illustration of crowdsourcing in the context of the three SPD variables.

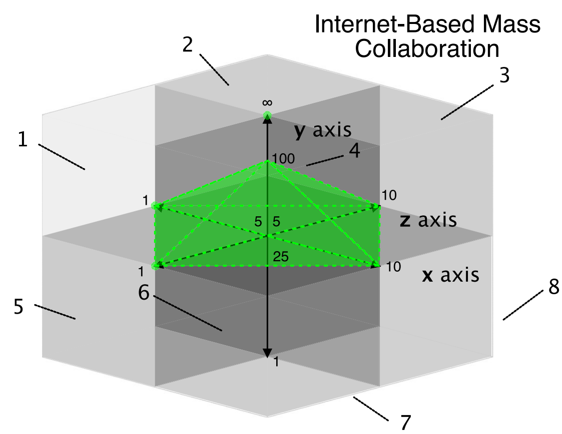

Fig. 7. Illustration of IBMC in the context of the three SPD variables.

phases, the user derives the values for N, A, and o representing their preferences and capabilities relating to their design problem. The user then maps these variables on the axis, as shown in Figs. 5–8, allowing their SPD quadrant to be revealed. The case study shown in Table VII demonstrates this process.

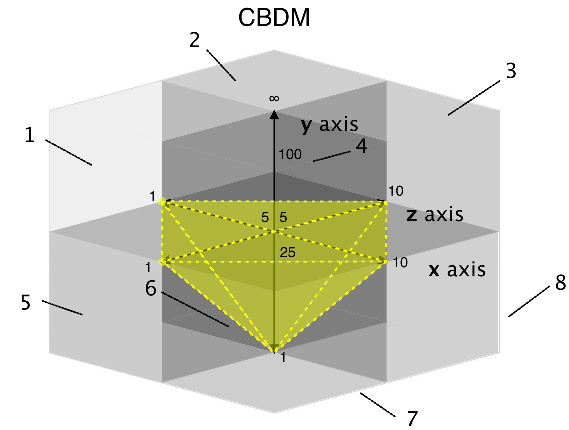

Fig. 8. Illustration of CBDM in the context of the three SPD variables.

TABLE V
SPD QUADRANT 1 TO 4

	Quadrant 1	Quadrant 2	Quadrant 3	Quadrant 4
Org Proximity	1 - 5	1 – 5	5 - 10	5 - 10
Actor Proximity	1 - 5	5 - 10	5 - 10	1 - 5
Number	25 - ∞	25 - ∞	25 - ∞	25 - ∞
Value Flow	External to Internal	Between external and then to Internal	Between external and then to internal	Familiar External to Internal
Description	Access to large and diverse crowd. Minimal collaboration between contributors.	Crowd work together to create solutions that are then provided to organisation. Project ownership is more decentralized	Crowd work together to create solutions but organisation is more involved. Project ownership is more central to	External parties provide value to organisation but do not collaborate with each other
			the organisation.	
Main Web 2.0 tool used	Social networking	Instant messaging	Instant messaging	Instant messaging, VoIP, file sharing
Also known as	Crowdsourcing	Wikipedia	Mass Collaboration	Large scale Outsourcing

TABLE VI
SPD QUADRANT 1–4

	Quadrant 5	Quadrant 6	Quadrant 7	Quadrant 8
Org Proximity	1 – 5	1 - 5	5 - 10	5 - 10
Actor proximity	1 - 5	5 - 10	5 - 10	1 - 5
Number	1 - 25	1 - 25	1 - 25	1 - 25
Value Flow	External to internal	Between external to internal	External to Internal to External	External to internal
Desc.	Value to the company is created by non-collaborative parties and then provided to the organisation	Value to the company is created in a community without the organisation's input, it is then provided/bought by the organisation (University IP)	Actors collaborate with each other and are close to the organisation	Expert actors provide and collaborate with the company but not with each other
Main Web 2.0 tool used	SaaS	SaaS	VoIP, instant messaging	VoIP, instant messaging
Also known as	Individual agencies/inventors	Think Tanks	Mass Collaboration	Small scale outsourcing

VI. Validation of the SPD Framework

Validation of engineering research has traditionally demanded formal, rigorous, and quantitative validation [48]. There are some areas of engineering research, however, that rely on subjective statements which make quantitative validation problematic [49]. The SPD framework presented in this article exists within this realm of engineering research. The authors, therefore, sought validation methods that offered rigorous and formal validation of design methods.

Seepersad [48] offer one such validation approach defined as the validation square. Fundamental to this approach "is a process of building confidence in its usefulness with respect to a purpose" [48] consisting of four validation stages. The first two stages examine the structural validity of the design method determining whether the construction process is both effective and efficient. The latter two stages examine the performance of the design method, determining whether the design method provides effective and efficient design solutions. How these validation stages are defined in the context of the validation of the SPD framework is shown in Table VIII.

TABLE VII
USING THE SPD FRAMEWORK AND TENANT SELECTION TOOL

SPD Design Phase	Description	Case Study Context
Task Clarification	What is the problem to be solved?	My in-house design team is struggling to present a novel concept for an off-road wheelchair design.
Requirements Specification	What do we need to solve this problem?	My team is lacking in diversity and have limited experience of using or designing wheel chairs. We are also a small team of five and so require a greater number of perspectives to generate ideas. We also have limited time available and need new ideas within the next few months.
	N	Greater than 5
	a	Less than 5
	o	Less than 5
Tenant Selection and Evaluation	Use the SPD Quadrants to determine which SPD tenant to use.	According to the definition model, SPD tenants with high number of participants that are detached from each other and the organisation in proximity includes crowdsourcing and Open Innovation.
	Make decision on which SPD initiative	Crowdsourcing allows ideas to be gained from a large number of people with varied perspectives in a relatively short amount time. Open Innovation, on the other hand, requires
		the construction of formalised knowledge sharing channels. Crowdsourcing is therefore most suitable for this case.
Detailed Form Design	Make decisions on the structure, performance and content of the chosen SPD tenant	In the case of crowdsourcing, due to time constraints a crowdsourcing contest will be promoted using existing platforms such as Facebook and Twitter. The contest will be open for two months, allowing the remaining time to be used for submission evaluation. The prize money will be calculated according to the available budget factoring in the cost to evaluate submissions.
Launch	Apply the SPD tenant	In the case of crowdsourcing, this stage is the launch of the crowdsourcing contest
Results Processing	Determine whether SPD tenant has produced appropriate an appropriate design solution	In the case of crowdsourcing, this process includes the evaluation of submitted ideas according to the defined quality criteria
Documentation	Document the process to reveal learning outcomes for future application	

TABLE VIII
VALIDATION SQUARE IN THE CONTEXT OF VALIDATION OF THE SPD FRAMEWORK

Validation Stage	Definition
Theoretical Structural Validity	Has a rigorous and methodical application of theory been used to construct the framework?
Empirical Structural Validity	Has the theory used to construct the framework been proven to offer effective design solutions?
Theoretical Performance Validity	Demonstrate that the design method can provide a design solution in an efficient way.
Empirical Performance Validity	Use the design method to demonstrate that the design method can provide effective design solutions.

A. Theoretical Structural Validity

The validity of the SPD framework, in this stage, is judged according to the process chosen to construct the SPD framework. In the absence of an existing construction method, the authors chose to use an existing design method. Existing design methods were chosen from a recent, "detailed and comprehensive" presentation of design methods, as provided by Wynn and Clarkson [36], and design methods were rejected over a series of detailed evaluation stages.

B. Empirical Structural Validity

The validity of the SPD framework, in this stage, is judged on the validity of the underpinning theory to provide useful design solutions. In this case, the design methods most similar to the SPD framework are prescriptive design methods such as Pahl and Beitz [39], Ullman's Stage Model [45], and VDI2221 [43]. Each of these design methods, as described by Wynn and Clarkson, "convey best practices intended to guide real-world situations" [36] and have been rigorously validated by the design community.

C. Theoretical Performance Validity

To demonstrate the theoretical performance validity of the SPD framework, a case study was presented in Table VII. This demonstrates the framework's use for the theoretical generation of a design solution.

VII. LIMITATIONS AND FUTURE WORK

The authors recognize three limitations with the existing framework that invites further research. First, the framework is at a high level of abstraction. This means it does not offer adequate guidance on the specific design decisions needed for individual tenant application. Second, the fourth stage of the validation square [48] is yet to be completed. Third, the framework requires a foundation of knowledge on SPD tenants.

The SPD framework is a high-level design framework to support the application of all SPD tenants, each with varying applications and dynamics. As a consequence, it does not effectively guide the specific decisions needed for individual tenants. In the case study, presented in Table VII, the detailed form design stage includes decisions specific to the design of crowdsourcing initiatives. These can only be made by those with a deeper knowledge of crowdsourcing initiative design, which is a significant limitation of the framework. Panchal [17] offers a framework for the design of crowdsourcing initiatives that could support the design phase of the SPD framework, but further work is required for other tenants. Ultimately, this framework needs supplementary material to be used outside of the realm of academia.

The fourth stage of validation is required, as part of future work, to prove the framework's empirical performance validity. In theoretical performance validation, the case study demonstrated its ability to produce a solution. In empirical performance validation, the framework should be used to produce a solution that is proven to be effective. This validation process begins by defining what effectiveness means, in the context of the generated solution [48].

The final limitation is the level of preexisting knowledge on SPD and existing motivation to use SPD, needed to apply the framework. As stated by Forbes and Schaefer [13], SPD "is a relatively undeveloped and unexplored term within both academia and the context of technology transfer to industry." This means that the information derived in the requirements phase of the case study may be idealistic. It is likely that practitioners seeking to apply the framework will require further education on SPD to apply the framework effectively. Further work is, therefore, required to ensure that this framework can be used without additional consultation.

VIII. CONCLUSION

The main contribution of this article was an SPD framework that guided the selection and design of an SPD tenant for application in a product development context.

In the absence of existing prescriptive literature on the designing of a design framework, the authors adapted an existing design method and present an adaptation of Pahl and Beitz [29], Ullman's Stage Model [33], and the VDI2221 Stage Model [43]. In the SPD framework, Concept generation and evaluation was instead tenant selection and evaluation and incorporates the use of a tenant selection and evaluation tool titled the quadrants of SPD. The derivation of this tool presented three variables that all SPD tenants can be defined by: the number of actors N, the proximity of the actor to the organizations o, and the proximity of the actors to each other a. The emergence of these common variables allowed the selection of an SPD tenant for application, according to the needs and capabilities of the user. A case study, demonstrating the use of this selection tool, was presented and represented one of the three validation phases included in this article. For further development of this SPD framework, existing limitations of the framework can be addressed. The detailed form design phase of the framework required design decisions regarding the SPD tenant to be made but the literature guiding these specific decision processes is currently lacking. This SPD framework, therefore, is a high-level design process for the application of SPD tenants and offers an indicator of future research directions that will aid the application of SPD tenants.

References

[1] A. Peterson and D. Schaefer, "Social product development: Introduction, overview, and current status," in *Product Development in the Socio-sphere*. Cham, Germany: Springer-Verlag, 2014, pp. 1–33.

[2] G. Cormode and B. Krishnamurthy, "Key differences between Web 1.0 and Web 2.0," *First Monday*, vol. 13, no. 6, pp. 2–6, 2008.

[3] E. Constantinides and S. J. Fountain, "Web 2.0: Conceptual foundations and marketing issues," *J. Direct, Data Digital Marketing Practice*, vol. 9, no. 3, pp. 231–244, 2008.

[4] S. J. Andriole, "Business impact of web 2.0 technologies," *Commun. ACM*, vol. 53, no. 12, pp. 67–79, 2010.

[5] L. Thames and D. Schaefer, "Software-defined cloud manufacturing for industry 4.0," *Procedia CIRP*, vol. 52, pp. 12–17, 2016.

[6] D. Wu, D. W. Rosen, J. H. Panchal, and D. Schaefer, "Understanding communication and collaboration in social product development through social network analysis," *J. Comput. Inf. Sci. Eng.*, vol. 16, no. 1, 2016, Art. no. 011001.

[7] M. Chui and J. Bughin, "The rise of the networked enterprise: Web 2.0 finds its payday," *McKinsey Quartlerly*, 2010.

[8] K. Abhari, E. Davidson, B. S. Xiao, and Davidson, "Taking open innovation to the next level: A conceptual model of social product development (SPD)," 2016. [Online]. Available: https://www.researchgate.net/profile/Kaveh_Abhari/publication/301899474_Taking_Open_Innovation_to_the_Next_Level_A_Conceptual_Model_of_Social_Product_Development_SPD/links/572c654108ae0a72918a4af4/Taking-Open-Innovation-to-the-Next-Level-A-Conceptual-Mode. [Accessed on: January 10, 2018].

[9] M. Bertoni, K. Chirumalla, and C. Johansson, "Social technologies for cross-functional product development: SWOT analysis and implications," in *Proc. 45th Hawaii Int. Conf. Syst. Sci.*, 2012, pp. 3918–3927.

[10] L. Huston and N. Sakkab, "Connect and develop," *Harvard Bus. Rev.*, vol. 84, no. 3, pp. 58–66, 2006.

[11] H. Salojärvi, L. M. Sainio, and A. Tarkiainen, "Organizational factors enhancing customer knowledge utilization in the management of key account relationship," *Ind. Marketing Manage.*, vol. 39, no. 8, pp. 1395–1402, 2010.

[12] V. Acha, A. Davies, M. Hodbay, and A. Salter, "Exploring the capital goods economy: Complex product systems in the U.K.," *Ind. Corporate Change*, vol. 13, no. 3, pp. 505–529, 2004.

[13] H. Forbes and D. Schaefer, "Social product development: The democratization of design, manufacture and innovation," *Procedia CIRP*, vol. 60, pp. 404–409, 2017.

[14] A. Larsson *et al.*, "Distributed team innovation: A framework for distributed product development," in *Proc. Int. Conf. Eng. Design*, 2003, pp. 321–322.

[15] M. Bartl, G. Jawecki, and P. Wiegandt, "Co-creation in new product development: Conceptual framework and application in the automotive industry," in *Proc. Conf. R&D Manage.*, 2010, pp. 6–12.

[16] X. J. Niu, S. F. Qin, J. Vines, R. Wong, and H. Lu, "Key crowdsourcing technologies for product design and development," *Int. J. Autom. Comput.*, vol. 16, pp. 1–15, 2018.

[17] J. H. Panchal, "Using crowds in engineering design: Towards a holistic framework," in *Proc. 20th Int. Conf. Eng. Des.*, 2015, pp. 041–050.

[18] S. Abrahamson, P. Ryder, and B. Unterbeg, *Crowdstorm: The Future of Innovation, Ideas, and Problem Solving*. New York, NY, USA: Wiley, 2013.

[19] E. Cullina, K. Conboy, and L. Morgan, "Choosing the right crowd: An iterative process for crowd specification in crowdsourcing initiatives," in *Proc. 49th Hawaii Int. Conf. IEEE Syst. Sci.*, 2016, pp. 4355–4364.

[20] R. J. Gerth, A. Burnap, and P. Papalambros, "Crowdsourcing: A primer and its implications for systems engineering," in *Proc. NDIA Ground Veh. Syst. Eng. Technol.*, 2012, pp. 1–10.

[21] A. Kittur, B. Smus, S. Khamkar, and R. E. Kraut, "Crowdforge: Crowdsourcing complex work," in *Proc. 24th Annu. ACM Symp. User Interface Softw. Technol. ACM*, 2011, pp. 43–52.

[22] H. Chesbrough, W. Vanhaverbeke, and J. West, *Open Innovation: Researching a New Paradigm*. London, U.K.: Oxford Univ. Press, 2006.

[23] M. Fathianathan, J. H. Panchal, and A. Y. C. Nee, "A platform for facilitating mass collaborative product realization," *CIRP Annals*, vol. 58, no. 1, pp. 127–130, 2009.

[24] B. Unterberg, "Crowdsourcing (Jeff Howe)," in *Social Media Handbuch*. Baden-Baden, Germany: Nomos Verlagsgesellschaft mbH & Co., 2010, pp. 121–136.

[25] H. Forbes and D. Schaefer, "Crowdsourcing in product development: Current state and future research directions," in *Proc. 15th Int. Des. Conf.*, 2018, pp. 579–588.

[26] M. Elliott, "A framework for understanding and designing mass collaboration," in *Mass Collaboration and Education*. Cham, Germany: Springer-Verlag, 2016, pp. 65–84.

[27] J. H. Panchal and M. Fathianathan, "Product realization in the Age of mass collaboration," in *Proc. Int. Des. Eng. Tech. Conf. Comput. Inf. Eng. Conf.*, 2008, pp. 219–229.

[28] D. Wu, J. L. Thames, D. W. Rosen, and D. Schaefer, "Towards a cloud-based design and manufacturing paradigm: Looking backward, looking forward," in *Proc. Int. Des. Eng. Tech. Conf. Comput. Inf. Eng. Conf.*, 2012, pp. 315–328.

[29] A. E. Stevenson, *Oxford Dictionary of English*. New York, NY, USA: Oxford Univ. Press, 2010.

[30] S. Mior, J. Barnsley, H. Boon, F. D. Ashbury, and R. Haig, "Designing a framework for the delivery of collaborative musculoskeletal care involving chiropractors and physicians in community-based primary care," *J. Interprofessional Care*, vol. 24, pp. 678–689, 2010.

[31] K. M. Boydell, N. Greenberg, and T. Volpe, "Designing a framework for the evaluation of paediatric telepsychiatry: A participatory approach," *J. Telemed. Telecare*, vol. 10, no. 3, pp. 165–169, 2004.

[32] F. Maon, A. Lindgreen, and V. Swaen, "Designing and implementing corporate social responsibility: An integrative framework grounded in theory and practice," *J. Bus. Ethics*, vol. 87, no. 1, pp. 71–89, 2009.

[33] T. Lowrie, "Designing a framework for problem posing: Young children generating open-ended tasks," *Contemporary Issues Early Childhood*, vol. 3, no. 3, pp. 354–364, 2002.

[34] J. Reeves, C. Forde, V. Casteel, and R. Lynas, "Developing a model of practice: Designing a framework for the professional development of school leaders and managers," *School Leadership Manage.*, vol. 18, no. 2, pp. 185–196, 1998.

[35] D. Wynn and J. Clarkson, "Models of designing," in *Design Process Improvement*, London, UK: Springer-Verlag, 2005, pp. 34–59.

[36] D. C. Wynn and P. J. Clarkson, "Process models in design and development," *Rese. Eng. Des.*, vol. 29, no. 2, pp. 161–202, 2018.

[37] David C. Wynn, C. M. Eckert, and P. J. Clarkson, "Guidelines for a decision support method adapted to NPD processes," in *Proc. 16th Int. Conf. Eng. Des.*, 2007, pp. 693–694.

[38] F. Harmsen and M. Saeki, "Comparison of four method engineering languages," in *Proc. Working Conf. Method Eng.*, 1996, pp. 209–231.

[39] G. Pahl and W. Beitz, *Engineering Design: A Systematic Approach*. New York, NY, USA: Springer-Verlag, 2013.

[40] J. H. Evans, "Basic design concepts," *J. Amer. Soc. Naval Eng.*, vol. 71, no. 4, pp. 671–678, 1959.

[41] W. Beitz and G. Pahl, "Engineering design: A systematic approach," MRS BULLETIN 71, 1996.

[42] M. J. French, J. T. Gravdahl, and M. J. French, *Conceptual Design for Engineers*. New York, NY, USA: Springer-Verlag, 1985.

[43] V. D. I. Richtlinie, *2221: Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte*. Düsseldorf, Germany: VDI-Verlag, 1993.

[44] V. Hubka and W. E. Eder, *Engineering Design*. Zürich, Switzerland: Heurista, 1992.

[45] G. D. Ullman, G. T. Deitterich, and A. L. Stauffer, "A model of the mechanical design process based on empirical data," *Artif. Intell. Eng. Des. Anal. Manuf.*, vol. 2, no. 1, pp. 33–52, 1988.

[46] B. López-Mesa and G. Thompson, "Exploring the need for an interactive software tool for the appropriate selection of design methods," in *Proc. 14th Int. Conf. Eng. Des.*, 2003, p. 7.

[47] T. Braun and U. Lindemann, "Supporting the selection, adaptation and application of methods in product development," in *Proc. 14th Int. Conf. Eng. Des.*, 2003.

[48] K. E. Lewis, W. Chen, and L. C. Schmidt, "The validation square: how does one verify and validate a design method?" in *Decision Making in Engineering Design*. New York, NY, USA: ASME Press, 2006, pp. 303–314.

[49] T. Munzner, "A nested model for visualization design and validation," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 6, pp. 921–928, Nov./Dec. 2009.

[50] N. Franke and F. Piller, "Value creation by toolkits for user innovation and design," *J. Product Innov. Manage.*, vol. 21, no. 6, pp. 401–415, 2004.

[51] S. Chamberlain, H. Sharp, and N. Maiden, "Towards a framework for integrating agile development and user-centred design," in *Proc. Int. Conf. Exterme Program. Agile Processes Softw. Eng.*, 2006, pp. 143–154.

[52] D. Schaefer, *Cloud-Based Design and Manufacturing (CBDM)*. New York, NY, USA: Springer-Verlag, 2014.

[53] D. Frydrych, A. J. Bock, T. Kinder, and B. Koeck, "Exploring entrepreneurial legitimacy in rewards-based crowdfunding," *Venture Capital*, vol. 16, no. 3, pp. 247–269, 2014.

[54] M. Bertoni, H. Eres, and J. Scanlan, "Co-Creation in complex supply chains: The benefits of a value driven design approach," in *Product Development in the Socio-Sphere*, Cham, Germany: Springer-Verlag, 2014, pp. 35–62.

[55] M. Bertoni *et al.*, "The rise of social product development," *Int. J. Netw. Virtual Org.*, vol. 11, no. 2, pp. 118–207, 2012.

[56] K. Gericke, J. Kramer, and C. Roschuni, "An exploratory study of the discovery and selection of design methods in practice," *J. Mech. Des.*, vol. 138, no. 10, 2016, Art. no. 101109.

[57] P. Le Masson, K. Dorst, and E. Subrahmanian, *Design Theory: History, State of the Art and Advancements*. New York, NY, USA: Springer-Verlag, 2013.

[58] J. Lindström, D. Plankina, H. Lideskog, M. Löfstrand, and L. Karlsson, "Functional product development: Criteria for selection of design methods on strategic and operational levels," in *The Philosopher's Stone for Sustainability*. New York, NY, USA: Springer-Verlag, 2013, pp. 25–30.

[59] Daniel Michelis and T. Schildhauer, *Social Media Handbuch: Theorien, Methoden, Modelle und Praxis*. Berlin, Germany; Nomos, 2010.

[60] W. Vanhaverbeke, "Rethinking open innovation beyond the innovation funnel," *Technol. Innov. Manage. Rev.*, vol. 3, no. 4, pp. 6–10, 2013.