A serious game for introducing set-based concurrent engineering in industrial practices

Endris Kerga[1], Monica Rossi[1], Marco Taisch[1] and Sergio Terzi[2]

Abstract

This article proposes a Serious Game about 'Set-Based Concurrent Engineering', which is one of the elements of lean practices in product design and development. Although Set-Based Concurrent Engineering is becoming popular in academia, in practice, understanding and adoption of it are low. Thus, the game presented in this article is designed to bring hands-on experience to practitioners to understand its principles and associated enablers. The game is structured in two stages, simulating the traditional approach to concept development, called 'Point-Based Concurrent Engineering', and the lean approach, called Set-Based Concurrent Engineering, respectively. Performance metrics are provided in the game to track teams' performances in the two stages. Several practitioners have played the game. This article also presents the feedback obtained from a game session to illustrate the educational purposes and effectiveness of the game.

Keywords

lean product development, set-based concurrent engineering, point-based concurrent engineering, serious gaming

Introduction

The traditional approach to developing a design concept typically starts with breaking it into its subsystems, defining detailed requirements for each module and deriving a small number of alternative solutions that appear to meet the initial requirements. Engineers then quickly assess the solutions and select the one option to be pursued. This process, however, rarely turns out to be linear because of the uncertainties in product development (PD) (Ward et al., 1995). These uncertainties can be caused by changes in customer requirements, failures during testing, issues in manufacturing manufacturability issues, compatibility failures and so on (Nahm et al., 2007; Oosterwal, 2010). A series of iterative loops follows to either modify the concept or select a completely different solution. Because of its iterative nature, where engineers move from point to point in the realm of searching for feasible solutions, this process is called 'Point-Based Concurrent Engineering' (PBCE) (Al-Ashaab et al., 2013; Sobek et al., 1999; Ward et al., 1995).

'Set-Based Concurrent Engineering (SBCE)' is an alternative approach to PBCE, and its first industrial application is reported in Toyota's PD (Sobek et al., 1999; Ward et al., 1995). Researches claim SBCE to improve efficiency and effectiveness in PD (Al-Ashaab et al., 2010; Morgan and Liker, 2006; Oosterwal, 2010; Sobek et al., 1999; Ward et al., 1995, 2007).

Sobek (1997) defines SBCE as when engineers and product designers 'reason, develop and communicate about sets of solutions in parallel and relatively independent'. The definition can be well understood through its three basic principles (Sobek et al., 1999): (1) 'Map the design space', or the principle of exploration, which aims to achieve a thorough understanding of the sets of design possibilities for the subsystems; (2) 'Integrate by intersection', or the principle of set-based communication, which ensures that subsystem solutions defined are workable/compatible with all functional groups involved and (3) 'Establish feasibility before commitment', or the principle of convergence, that allows the aggressive elimination of inferior design solutions from sets and guarantees the arrival of high-value system solutions.

[1]Politecnico di Milano, Milan, Italy
[2]Università degli studi di Bergamo, Bergamo, Italy

Corresponding author:
Endris Kerga, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milan, Italy.
Email: endristemam.kerga@polimi.it

Researchers argue that SBCE has several advantages compared with PBCE (Kennedy and Harmon, 2008; Khan et al., 2011; Raudberget, 2010; Sobek, 1997; Takai, 2010; Ward et al., 1995). The advantages are as follows: (1) the reduction of product and process costs by searching for cheaper alternatives, adding the right features and reducing late/costly reworks; (2) the reduction of development time by reusing previous knowledge and/or avoiding delays caused by late changes; (3) the improvement of innovation potential by exploring alternative solutions and making ideas successful the first time; (4) the establishment of better communication by effectively using data and proven knowledge among design teams (e.g. trade-off and limit curves); (5) the avoidance of design risks by increasing the probability of success because of considering larger sets, front-loading previous knowledge and establishing feasibility before commitment and (6) the facilitation of learning through extensive testing and visually depicting lessons learned in trade-off and limit curves to simplify future reuse.

Although SBCE's principles are sound and the performance benefits are promising, an awareness gap hinders SBCE's wider acceptance in industries. Researchers have shown that the principles and the enablers of SBCE and the implementation procedure are unclear to industries. For instance, Bernstein (1998) conducted multiple case studies in the US aerospace industry. He observed that some principles and enablers of SBCE are applied in some of the aerospace companies, and some performance benefits are exhibited. However, Bernstein (1998) also reported that in many aerospace companies, the application of set-based thinking is very limited, and there is a low awareness level across the industry. A recent study conducted in Swedish companies showed similar results: SBCE is not well understood and is not implemented (Raudberget, 2010). Some of the companies studied obtained the following performance improvements: up to a 75% reduction in product cost, a 50% reduction in lead time, a 50%–75% improvement in product technical performances and a 50%–100% reduction in engineering reworks (Raudberget, 2010). To the contrary, other companies are observed to have negative results in applying SBCE. For example, in some companies, there was a 25% increment in lead time and a 25% increment in development costs. Some firms obtained neither gains nor losses in adopting SBCE. Those companies that had negative results and those that had zero loss/gain were asked about the reason for the results. Most companies answered that 'SBCE is not the way they normally used to work' (Raudberget, 2010). Similarly, in studies conducted on several manufacturing companies, it is reported that the awareness level of SBCE's principles as well as how to effectively apply SBCE in the PD process is not well understood yet (Hoppmann, 2009; Rossi et al., 2012).

From the aforementioned research studies, one can understand that SBCE shows benefits when engineers/managers are equipped with the understanding of its principles and enablers. Otherwise, the effort made to implement the principles and enablers of SBCE becomes inefficient and ineffective.

The low awareness level of SBCE across companies is a gap that needs further investigation. A method of learning how to execute the SBCE process is needed to introduce SBCE and provide hands-on experience to the practitioners. Such a method can make the theoretical principles of SBCE more tangible or concrete for practitioners (product designers and project managers), so those practitioners can reflect on SBCE's applicability before adopting it in real design projects. Therefore, the article proposes a Serious Game (SG), called an SBCE game, designed to educate industrial players about SBCE principles and SBCE's associated enablers.

The remaining parts of the article are divided in four sections. In section 'Serious games in product development', an introduction about gaming and a review of existing games in PD are presented. In section 'SBCE game', a detailed introduction is presented about the SBCE game developed and the SBCE enablers embedded. In section 'Discussion', the results of one game session are illustrated. This session focuses on the results of the game played by designers and managers working in a company that designs and develops innovative humidification and air-conditioning products. In section 'Conclusion', the conclusions of the article and future possible research are outlined.

SGs in PD

'Game' can cover a variety of forms, and its definition strongly depends on authors' perspectives of a game. SG is still not a well-defined term, and there are similar terms in the literature, such as 'Simulation Game', 'Game-Based Learning', 'Educational Game' and 'Edutainment'. In general, the application of a game aimed at education and learning can be defined as a 'Serious Game' (Wouters et al., 2011). Some researchers define SG as an activity whose main purpose is people learning about serious contexts through playing (Charsky, 2010; Zyda, 2005). SGs' main objectives are learning and education via games rather than pure entertainment (Egenfeldt, 2006). In SGs, players assume different roles and are involved in simple and complicated decision-making processes.

Existing SGs in PD

The application of SGs in the manufacturing domain (e.g. PD) is gaining wider attention (Pourabdollahian et al., 2012). At least three existing games in PD can be found in the literature: the Concurrent Engineering Simulation Game, COSIGA (Riedel and Pawar, 2001); GLOTRAN ('Training engineers for mastering new requirements in globally distributed manufacturing') (Hoheisel et al., 2000) and the New Product Development (NPD) simulation game, CityCar (Cousens et al., 2009)).

The first game is designed to promote parallel, foresighted and co-operative working in a distributed and Concurrent Engineering (CE) environment. Players interact in a PD scenario where they must do specific tasks and experience direct feedback as a result of their actions. The second game is a computer- and reality-simulation game addressing different organisational forms of PD and production. It simulates the development and manufacturing of a product in a distributed team setting. Players improve their communication and collaborative skills, and learn how to use related Information and Communication Technologies (ICT) tools. The third game has aims similar to those of the first two. Players represent the marketing, development, production and logistic operations in game plays. They learn the importance of teamwork, connectedness and collaboration in launching products successfully.

To compare the SBCE game with the others, it is important to underline the key objectives addressed in the games. The SBCE game advances the learning objectives and experiences of the existing games in PD. Generally, the three games focus on soft skills to educate about CE in distributed development scenarios. The importance of considering alternative sets at the early phases of a design process is not considered in the others. The challenges encountered and the tools/skills required to converge into an optimal design solution are not mediated in the existing games. Moreover, those games relay their PD process as a *Design-Build-Test* paradigm (i.e. making the decision first and then testing a design to receive feedback), which is a typical approach in PBCE (Kennedy and Harmon, 2008). The SBCE game, however, advances the learning objectives to encompass the introduction of how to explore alternative sets, the facilitation of communication about sets among design teams and the progressive convergence of sets to optimality. In this way, the game educates players about a lean process of a *Test-Build-Design* paradigm (i.e. delay decision making until enough knowledge is gathered) (Kennedy and Harmon, 2008; Ward et al., 1995).

Effectiveness of using SG to educate about SBCE

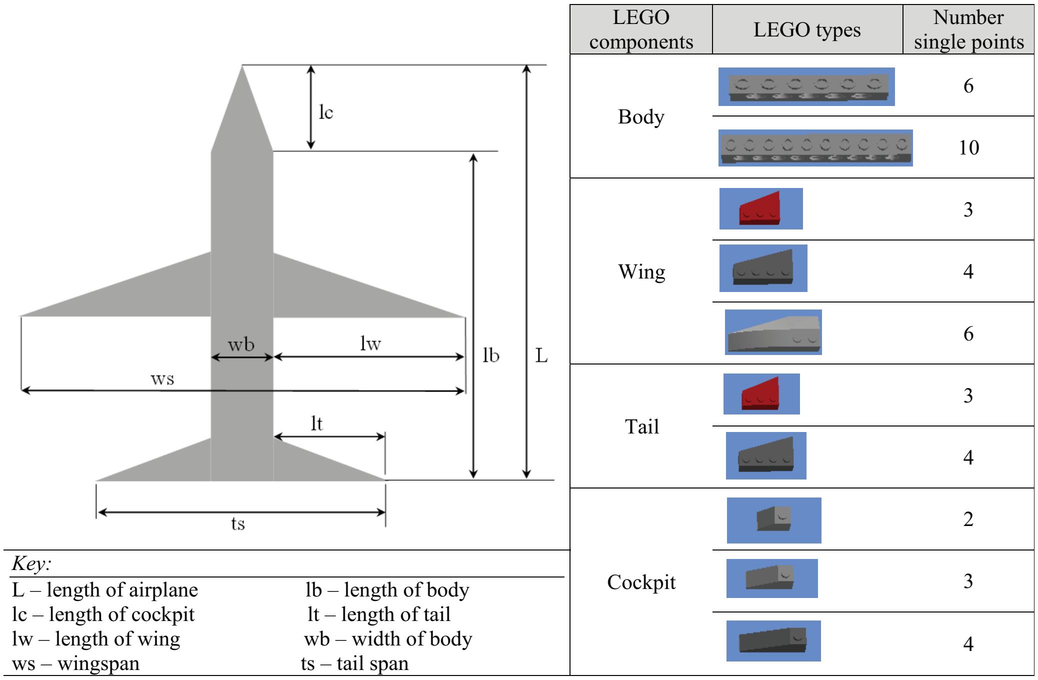
Six characteristics make an SG an effective mechanism in introducing new and complex practices as SBCE: (1) *internalise knowledge without interfering in an actual practice* – SGs put boundaries between actions in games and consequences in reality, but players acquire new skills and knowledge transferable to actual practices (Prensky, 2001); (2) *improve communication* – gaming creates a means to support effective communication and structures debates between stakeholders (Geurts and Joldersma, 2001); (3) *create consensus* – beyond communication, gaming creates means reaching consensus, conflict mediation and collaboration between actors' perceptions about a subject matter (Duke and Geurts, 2004); (4) *commitment to action* – gaming is used to introduce and test new concepts, to convince industrial players of the need for intervention, to introduce approaches to the intervention and to introduce the roles of the participants in the intervention process (Mayer, 2009); (5) *stimulate creativity* – gaming allows players to leave their routines and provides settings to experiment new ideas (Duke and Geurts, 2004) and (6) *understand complexity* – reality is much more complex than any attribution in gaming. For example, PD involves problems in a dynamic situation, with many variables, actors, objectives and uncertain outcomes. However, gaming creates a simulated and holistic environment to show how to make better decisions (Duke and Geurts, 2004).

Applying SBCE's principles and enablers is complex in practice. The phases involved in exploring alternative sets, communicating sets among subsystem functions, testing sets at different conditions and converging into an optimal system require changes in design paradigm, management approaches and leadership styles. Nonetheless, the changes require considerable amount of time to acquire skills. The characteristics of gaming, therefore, make SBCE a viable candidate to be simulated in an SG environment. Thus, industrial players can use the game as a first step in the change process to introduce, build consensus and identify the benefits and challenges related to SBCE.

SBCE game

In the SBCE game, players must design a simplified airplane structure, as shown in Figure 1, using different types of LEGO bricks. The aim of the players is to define the parameters of the airplane structure to satisfy given customer and technical requirements. The airplane structure has four subsystems to be designed (body, wing, cockpit and tail). The game is played with four players in a team, and each player represents a subsystem department.

Figure 2 shows the inputs and the stages of the game. The main inputs utilised to begin the game are customer requirements and the supplier catalogue (LEGO components). Having the inputs, players go through two main stages: Stage 1, where players design airplanes for



Key:

L – length of airplane
lc – length of cockpit
lw – length of wing
ws – wingspan
lb – length of body
lt – length of tail
wb – width of body
ts – tail span

LEGO components	LEGO types	Number single points
Body		6
		10
Wing		3
		4
		6
Tail		3
		4
Cockpit		2
		3
		4

Figure 1. A simplified airplane structure, parameters to be designed and LEGO bricks to use.

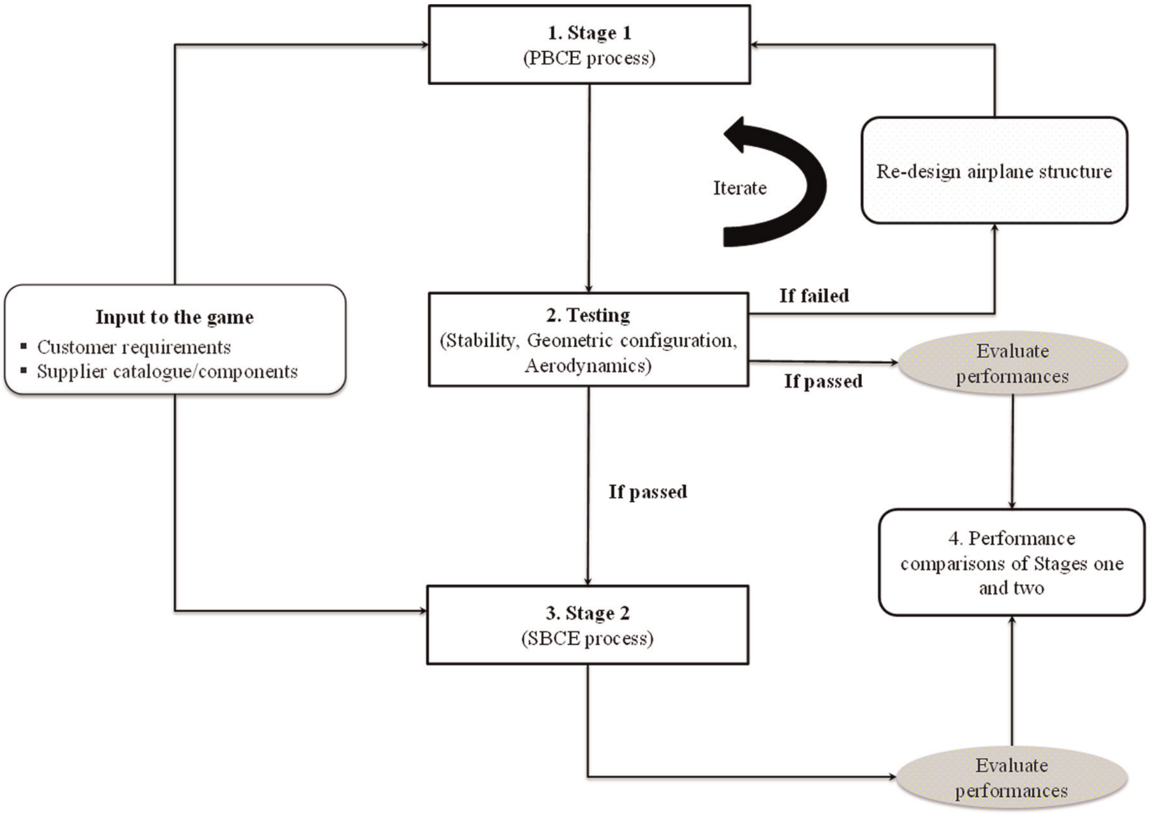


Figure 2. Inputs, steps and evaluation of learning outcomes.
PBCE: Point-Based Concurrent Engineering; SBCE: Set-Based Concurrent Engineering.

Group 1		
Customer Requirements		
Ranges	Min	Max
Number of passengers	91	110
Length of airplane	10	22
Weight of airplane	9,500	14,500
Wingspan	7	20
Tail span	7	15

Figure 3. An example of customer requirements.[1]

a given list of customer requirements following PBCE processes, and Stage 2, where players are provided with the necessary instruments to execute the SBCE processes to satisfy a given list of customer requirements. When players design their prototypes in Stage 1, they have to pass the testing constraints provided by a game facilitator. After Stages 1 and 2, a breakdown of the performances of players is provided to reflect on the differences between following PBCE and SBCE.

The following subsections are dedicated to discussing in depth the inputs, the steps and the assumptions made in designing the SBCE game.

Customer requirements

In the game, players must satisfy five customer requirements, as shown in Figure 3: number of passengers (*Np*); length of airplane (*L*); weight of airplane (*W*), which is the sum of the weight of the airplane structure (W_a) and the total passengers' weight (W_p); wingspan (*ws*) and tail span (*ts*). Each team acts as a 'company' that develops an airplane structure for a particular customer. A team has to design an airplane structure that will be in the ranges of the customer requirements.

The customer requirements are made intentionally vague. For example, the *Np* might be from 91 to 110 passengers, and the *ws* could be 7–15 distance units. Range-based requirements reflect the reality in which customers often suggest imprecise information and force designers to explore conceptual solutions wide open (Zhang et al., 2013). In the game, these requirements can be handled in different ways in Stage 1 (the PBCE process) and Stage 2 (the SBCE process). In the former case, players pick single points in the ranges of the requirements and proceed to define the design. This decision, however, results in compatibility and feasibility issues in the process. Moreover, players have less visibility on the possible subsystem solutions. While players do the SBCE process, a whole range of requirements is considered and possible subsystem solutions are explored. Thus, players understand the advantage of following the SBCE rather than the PBCE process to correctly fulfil the requirements the first time and have visibility on possible solutions.

LEGO components

The supplier components are LEGO bricks in different sizes and shapes that are used to build the subsystems (see Figure 1). Each brick has circular points on the top, and the numbers of points on the top of a brick define the characteristics of the component. A single point on a brick has the following attributes: cost (10 cost units), lead time or component ordering time (0.5 time unit), capacity (3 passengers), weight (100 weight units) and diameter (1 distance unit).

The aims of players are to design airplanes using the LEGO characteristics. The capacity, weight and diameter attributes are used to define the parameters of the airplanes and are in the range of the customer requirements. The attributes of cost and lead time are used to determine the total component cost and the total ordering time of an airplane.

Stage 1 and testing

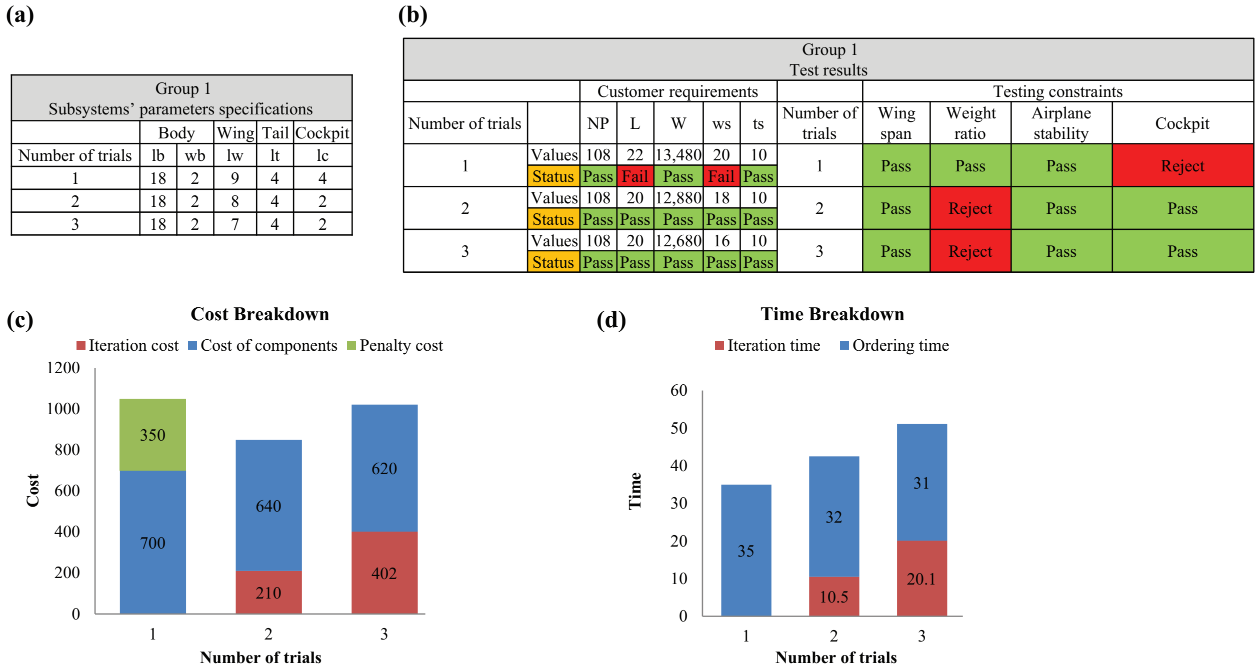
Taking the customer requirements and the supplier components, players are asked to build their airplanes in this stage. This stage simulates PBCE, where a team first designs an airplane structure and then tests it for customer requirements (shown in Figure 3) and testing constraints (listed in Table 1). The *Design-Build-Test* approach is what many non-lean companies follow at an early stage of design (Kennedy and Harmon, 2008). In Stage 1, players go through a trial-and-error approach to find an airplane that satisfies customer requirements and testing constraints. If a solution cannot fulfil a customer requirement or pass a testing constraint in a trial, a team should modify or define another solution in the next trial. However, redesigning has cost penalties and time penalties. The game facilitator acts as the testing department. The facilitator of the game acts as a testing department.

For example, Figure 4 shows the airplane specifications and the associated test results of the three trials of a team designing an airplane for the customer requirements given in Figure 3. Figure 4(a) shows the parameters of each subsystem the team has defined in each trial. Figure 4(b) shows the test results for each trial. For example, in the first trial, the team did not pass two of the customer requirements *ws* and *L* and the fourth testing constraint (see Table 1). Thus, the team needs to redesign the airplane and go to the second trial.

Stage 1 ends whenever a team satisfies all the requirements and passes all the testing constraints or

Table 1. Assumed passing constraints.

	Constraints type	Constraints equation
1.	Geometric configuration of wingspan	${}^{2}\!/_{3}L \leq ws \leq L$
2.	Weight ratio (R_w)	$R_w = \dfrac{W_p}{W_a} < 1.25$, where W_a is the weight of an airplane structure and W_p is the total weight of the passengers in an airplane. Each passenger is assumed to be 60 units of weight
3.	Airplane stability	$lt < lw$
4.	Alignment between body and cockpit	$lc = wb$



(a)

Group 1 Subsystems' parameters specifications					
	Body		Wing	Tail	Cockpit
Number of trials	lb	wb	lw	lt	lc
1	18	2	9	4	4
2	18	2	8	4	2
3	18	2	7	4	2

(b)

Group 1 Test results											
		Customer requirements					Testing constraints				
Number of trials		NP	L	W	ws	ts	Number of trials	Wing span	Weight ratio	Airplane stability	Cockpit
1	Values	108	22	13,480	20	10	1	Pass	Pass	Pass	Reject
	Status	Pass	Fail	Pass	Fail	Pass					
2	Values	108	20	12,880	18	10	2	Pass	Reject	Pass	Pass
	Status	Pass	Pass	Pass	Pass	Pass					
3	Values	108	20	12,680	16	10	3	Pass	Reject	Pass	Pass
	Status	Pass	Pass	Pass	Pass	Pass					

Figure 4. An example of airplane specifications of three trials: (a) associated test results, (b) team performances, (c) development cost and (d) time.

when the time allotted to Stage 1 expires (a maximum of 2 h is allowed for this stage).

Players are not supported by the game facilitator's telling them what process to follow at this stage. This approach was decided on to simulate how industrial practitioners execute the early phase of design. Often, even professional designers and engineers tend to follow PBCE and proceed with a trial-and-error paradigm. However, members of a team are left to discuss concerning any decision related to defining their airplane's structure – creating a concurrent environment.

Performance evaluations

After Stage 1, players are given a breakdown of their performance in terms of cost and time. The determinations of development cost and time are given in Tables 2 and 3.

Total development cost has three metrics: cost of components, cost of iteration and cost of penalty. Total development time has two metrics: ordering time and iteration time. These performance metrics are used for two purposes. First, they motivate teams. While teams compete to achieve better performances, the metrics are used to follow up on their achievements. Second, these metrics educate players about the impact of their decisions on PD performance.

Cost of iteration (or cost of rework), iteration time (or rework time) and penalty cost are proportional to the number of iterations players made because of failures in meeting either a customer requirement or/and a testing constraint. These metrics are used in the game

Table 2. Rules to measure total development cost.

Total development cost (C)

$C = \textit{cost of components}(cc) + \textit{cost of iteration}(ci) + \textit{cost of penality}(cp)$, where $cc = \textit{total number of points in an airplane} \times \textit{single point cost}$, $ci = 30\% \times cc$ (this is an additional cost if a prototype airplane fails to pass a testing constraint), and cp is an additional cost if a team fails to meet a customer requirement. The cp is determined based on unsatisfied customer requirements, and applying the following rules:

Unsatisfied customer requirements	Np	L	ws	ts	W
c_p	$30\% \times cc$	$40\% \times cc$	$10\% \times cc$	$5\% \times cc$	$20\% \times cc$

The calculations are pure assumptions used as rules in the game; they might not reflect the real situation in aircraft development.

Table 3. Rules to measure total development time.

Total development time (T)

$T = \textit{total ordering time}(tot) + \textit{iteration time}(it)$, where $tot = \textit{total number of points in an airplane} \times \textit{single point lead time}$, $it = 30\% \times tot$ (this is an additional time a team will be penalised if a prototype airplane fails to pass a testing constraint and misses a customer requirement).

Playing time (design time) is not considered in the calculation because players spent a considerable amount of time understanding the game itself. Thus, considering this time biases the outcomes. Therefore, design time is considered to be the same for all teams (2 h), and players are evaluated based on the time and cost measures given previously.

to educate players that PBCE has a disadvantage in assuring success during testing. In PBCE, because players consider a single solution at a time, the probability of success in passing customer and testing constraints is low. As a result, success is not guaranteed to be as early as possible. For instance, as shown in Figure 4(b), the team keeps failing to meet customer requirements and testing constraints in different trials. Because of these failures, the costs of iteration and penalty as well as those of iteration time increase.

Cost of components and ordering time are proportional to the number of LEGO components players embedded in their airplane designs. As shown in Figure 4(c) and (d), for example, the team improves these metrics as trial-and-error bases. In PBCE, because there is not visibility in the alternatives to consider, players have to design and test solutions one at a time to obtain the optimal solution. Moreover, as a result of not exploring possible solutions early, the team cannot guarantee if a solution is the best design that has the minimum possible numbers of LEGO components.

Stage 2 and supporting enablers

Before Stage 2 begins, discussions about the PBCE process are presented. Players and the facilitator(s) identify key shortcomings encountered in the first stage. The reasons for high development costs and time are considered as topics for discussion. For instance, 'What are the reasons for a high number of iterations and what are the associated impacts on costs and delivery time?' 'Why did players miss achieving customer requirements and what are the associated penalty costs?' and 'Why did players add unnecessary components and what are the impacts on cost and time?' are some of the questions posed to participants and are topics for the briefing. Throughout the discussions, the importance of lean concepts in delivering the right product at the right time and cost is stressed. Furthermore, introductions to the SBCE's principles and their benefits in overcoming the challenges faced in the first stage are discussed.

In the second stage, players follow a structured SBCE process. This stage simulates an approach different from the first. Here, players follow a *Test-Build-Design* paradigm, and design decisions are made as late as possible until feasibilities are proven. In summary, players responsible for different subsystems do the following: first, explore the possible sets of body, wing, cockpit and tail; second, communicate about the sets to eliminate incompatible subsystem solutions; third, evaluate the feasibilities of the sets of airplane combinations by front-loading the knowledge from testing constraints; and finally, evaluate feasible combinations against objective criteria, such as the total development time and cost. Figure 5 shows the four steps players should follow. Below, each step is explained.

Exploration of alternative subsystems. At this step, players are supported by the Quality Function Deployment (QFD) tool to explore alternative subsystem solutions and to be able to integrate customer requirements into

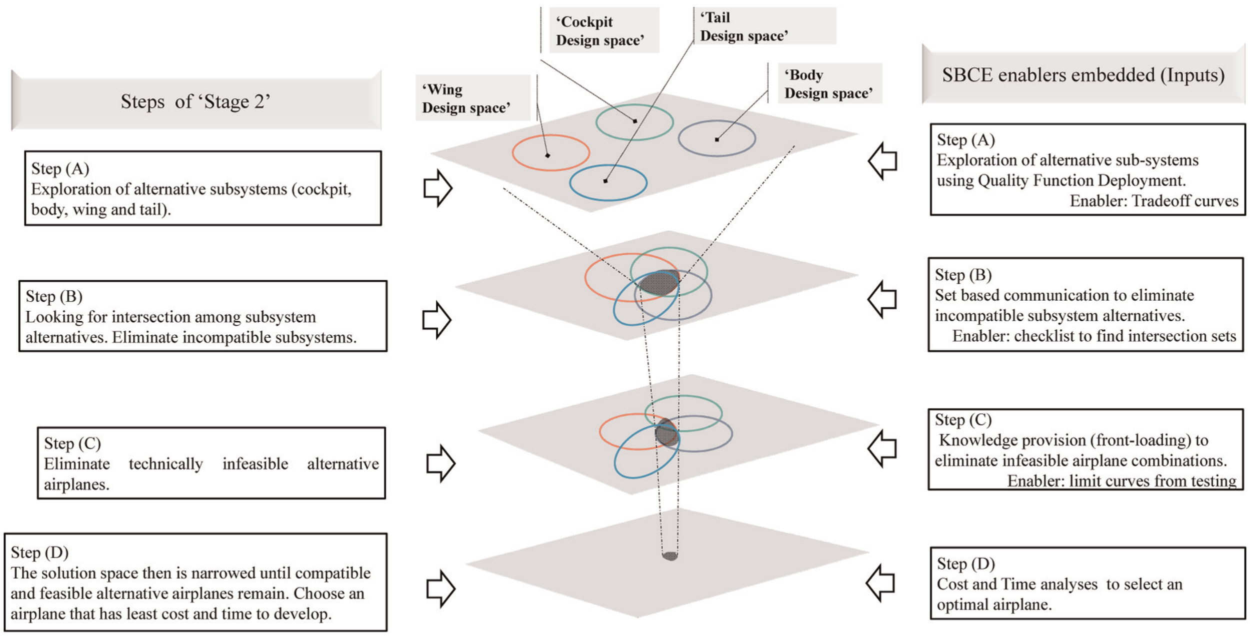


Figure 5. Stage 2 – steps and enablers.
SBCE: Set-Based Concurrent Engineering.

possible airplane parameters. QFD is a powerful tool in applying the SBCE process; it helps designers translate rough customer requirements into alternative subsystem solutions (Liker et al., 1996).

Therefore, each player responsible for a subsystem is given its own QFD chart in the form of a trade-off curve. SBCE research encourages using trade-off curves not only to explore but also to visually depict possible solutions and to simplify the pictorial representation of the governing knowledge (Sobek et al., 1999; Ward et al., 1995, 2007).

For example, in Figure 3, where the customer requirement for *Np* is (91, 110) passengers, the body department explores all the possible body modules in terms of the sizes (*wb* and *lb*) that can satisfy the customer within the range. In Figure 6, the trade-off curve provided for the body department is used to map possible body modules. For *Np* (91, 110), the possible body modules (*wb* and *lb*) are (3, 12), (2, 16) and (2, 18). These body sizes are the possible solutions to satisfy the customer requirement of *Np* (91, 110). Similarly, the cockpit department needs to map the possible body modules (*wb* and *lb*) that can satisfy the customer requirement *L*. For example, taking *L* (10, 22) from Figure 3 and using the trade-off curve provided for the cockpit department (see Figure 6), possible body modules can be defined from the cockpit department perspective. In this case, the possible body modules (*wb* and *lb*) are (2, 10), (2, 12), (2, 16), (2, 18), (3, 10), (3, 12), (3, 16), (3, 18), (4, 10), (4, 12) and (4, 16). These body sizes are possible solutions to satisfy the customer requirement of L (10, 22).

In sum, using the trade-off curves provided for the subsystems departments, possible solutions are mapped out (possible sizes for body, cockpit, wing and tail). The main educational purpose of this step is to let players understand how to relate customer requirements given in ranges to possible design solutions. Moreover, this step has the importance of educating players about the use of trade-off curves in the SBCE process for the effective elimination of subsystem solutions that cannot meet customer requirements.

Elimination of incompatible subsystems. Players at this step can eliminate incompatible subsystem solutions. Communication based on sets is discussed as a key step in the SBCE process (Inoue et al., 2013; Liker et al., 1996; Sobek et al., 1999). This helps define workable solutions for the teams involved. In practice, this step can be facilitated by simple checklists, physical tests or using the knowledge of experienced designers (Sobek et al., 1999). In this step of the game, players are supported by simple checklists that enable teams to define compatible subsystem modules. For example, in Step A, the player in the body department defines possible body sizes to meet the requested *Np*, and the other player in the cockpit department defines sizes that are acceptable in meeting the requested *L*. These two players, however, need to communicate to eliminate those body modules that are not in common.

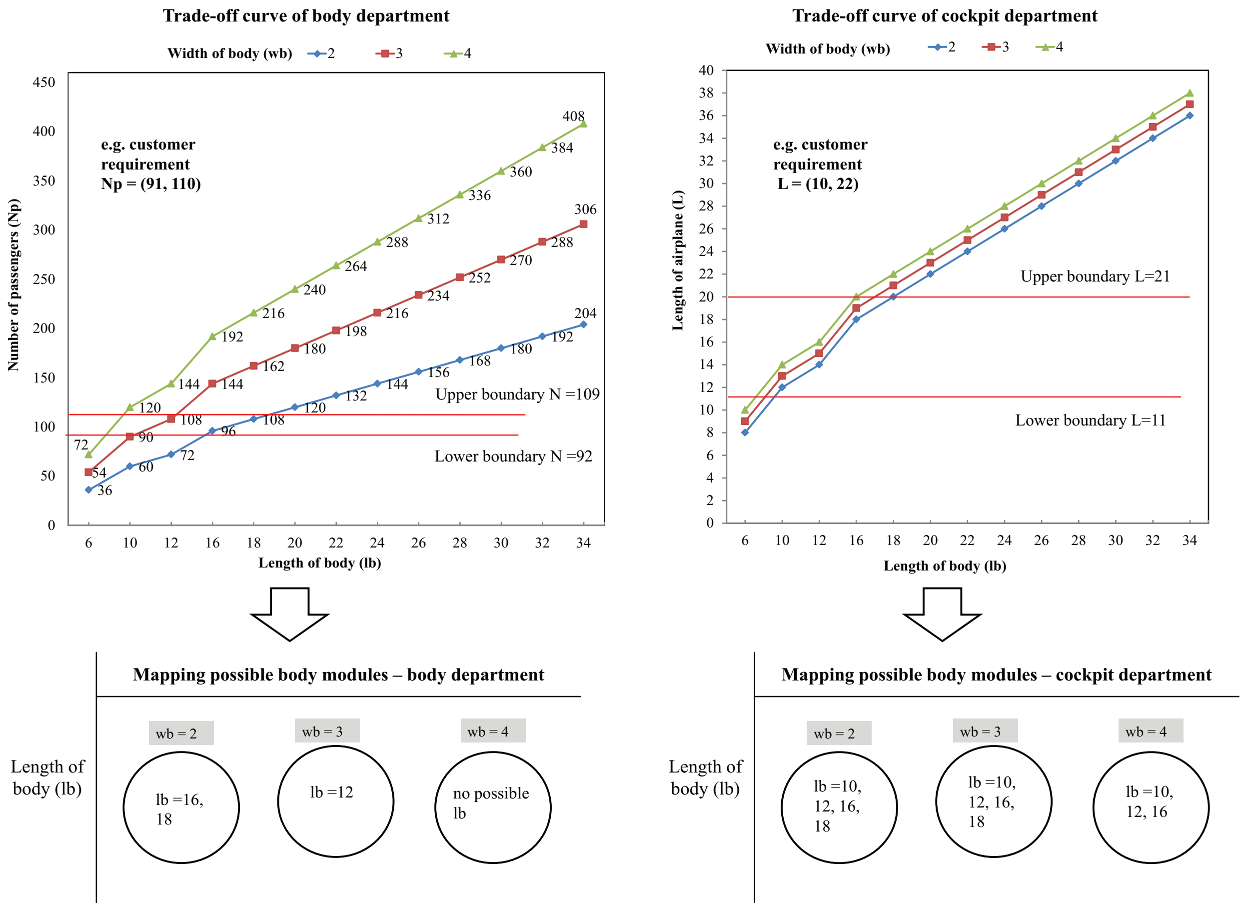


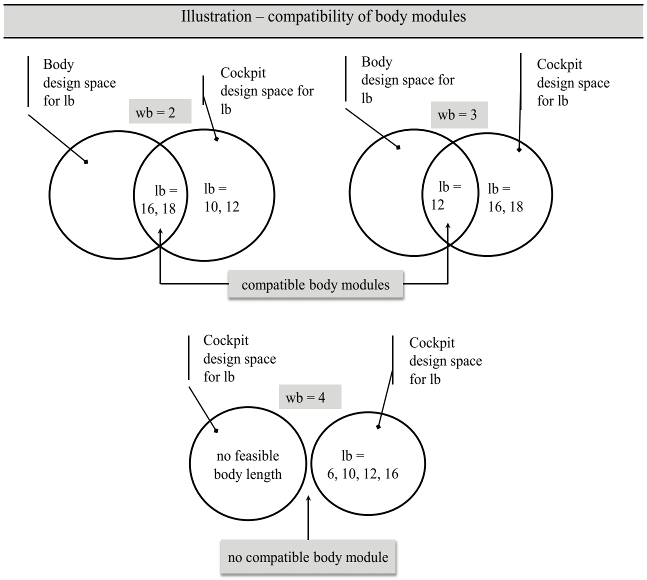
Figure 6. Mapping possible body modules from body and cockpit departments' perspectives (for Np = (91, 110) and L = (10, 22)).

Take the body modules explored in Step A from both the body and cockpit perspectives (see Figure 6). To eliminate incompatible solutions, the body and cockpit departments use the checklist provided in the game to eliminate incompatible body modules, as shown in Figure 7. In this example, the only body modules (*wb* and *lb*) that are workable for both the body and cockpit departments are (3, 12), (2, 16) and (2, 18). The rest of the solutions are eliminated from the solutions space.

The main educational purpose of this step is to allow players to understand that one elimination aspect in SBCE is based on compatibility. Since most systems are interrelated, before committing to a design solution subsystem, teams need to rule out incompatible subsystems by using simple compatibility checklists.

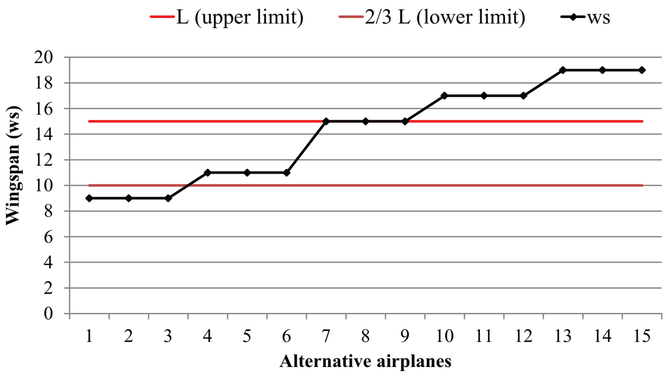
Elimination of infeasible airplanes. In Step B, players have complete alternative and compatible airplane combinations, but need to filter those using physical constraints. Physical constraints are paramount in the SBCE process to aggressively eliminate infeasible solutions. In practice, constraints or sources of failures might not be known at an early phase of a new PD process. But simulations, physical tests and previous knowledge can be used to front-load the process and evaluate sets for convergence. Moreover, depicting constraints as limit curves is key in the SBCE process to generalise and visually represent technical knowledge (Oosterwal, 2010; Ward et al., 2007). In the game, the physical constraints come from the testing department and are listed in Table 1. Therefore, in this step, players eliminate among those solutions defined in step C those that cannot pass all the testing constraints listed in Table 1. See Figure 8 for an illustrative example of a limit curve used to eliminate infeasible airplane solutions in the game.

Selection of an optimal airplane. When the alternative feasible airplanes are identified in Step C, estimating the cost and development time of each solution is used to select an optimal airplane solution. As referenced by



	Alternative body modules		Customer requirements				
	lb	wb	Number of passengers Np = (91, 110)		Length of airplane L = (10, 22)		Compatibility
			value	ok/no?	value	ok/no?	
Explored by body department	2	16	96	ok	16	ok	compatible
	2	18	108	ok	20	ok	compatible
	3	12	108	ok	15	ok	compatible
Explored by cockpit department	2	10	60	no	12	ok	incompatible
	2	12	72	no	14	ok	incompatible
	2	16	96	ok	18	ok	compatible
	2	18	108	ok	20	ok	compatible
	3	10	90	no	13	ok	incompatible
	3	12	108	ok	15	ok	compatible
	3	16	144	no	17	ok	incompatible
	3	18	162	no	19	ok	incompatible
	4	10	120	no	14	ok	incompatible
	4	12	144	no	16	ok	incompatible
	4	16	192	no	20	ok	incompatible

Figure 7. Checklists for communication between the body and cockpit departments.



Alternative airplanes	lb	wb	lw	lt	lc	Feasibility
1	12	3	3	3	3	Fail
2	12	3	3	4	3	Fail
3	12	3	3	6	3	Fail
4	12	3	4	3	3	Pass
5	12	3	4	4	3	Pass
6	12	3	4	6	3	Pass
7	12	3	6	3	3	Pass
8	12	3	6	4	3	Pass
9	12	3	6	6	3	Pass
10	12	3	7	3	3	Fail
11	12	3	7	4	3	Fail
12	12	3	7	6	3	Fail
13	12	3	8	3	3	Fail
14	12	3	8	4	3	Fail
15	12	3	8	6	3	Fail

Figure 8. Example of a limit curve.

Tables 2 and 3, in second stage, ci and it are zero, because all the causes of failures are anticipated in Step C and no design iteration is encountered. Similarly, since all the ranges of the customer requirements are deployed in the exploration phase (Step A), the cost of a penalty (cp) is zero. Although these assumptions are taking an ideal situation in an SBCE application, in practice, SBCE helps to drastically reduce unnecessary late design reworks and missing customer targets (Raudberget, 2010). Taking the example in Figure 8, Table 4 shows the development cost and time of the feasible airplanes. Airplane No. 4 has the minimum development cost, but airplane No. 7 has the minimum development time. The selection in this case depends on the team's choice. If the team follows a cost-based strategy (minimum cost), airplane No. 4 would be chosen. If team follows a time-based strategy (i.e. fast development time), airplane No. 4 would be chosen. Otherwise, the team can ask the game facilitator, who acts as a customer, to rank the relative importance of cost versus time, and team can use this information to choose between the airplanes using the expected value (aggregated value of cost and time).

Stage 2 is a partially automated in MS Excel 2010 program for Steps C and D. It takes about 1.5 h to complete. In summary, the second stage is to lead players through stepwise phases of the SBCE process. The objective is to educate players on how to delay

Table 4. Development cost and time of feasible airplanes.

						Total development	
Feasible airplanes	*lb*	*Wb*	*lw*	*Lt*	*Lc*	Cost	Time
4	12	3	4	3	3	590	70.8
5	12	3	4	4	3	610	73.2
6	12	3	4	6	3	650	78
7	12	3	6	3	3	630	69.3
8	12	3	6	4	3	650	71.5
9	12	3	6	6	3	690	75.9

Table 5. Results in Stages 1 and 2 – an example from a game session.

	Stage 1			Stage 2			Percentage improvement (Stage 2 vs Stage 1)	
Team number	Number of trials	Total development time	Total development cost	Number of trials	Total development time	Total development cost	Total development time	Total development cost
1	3	120	1200	1	58	580	> 50%	> 50%
2	1	45.6	380	1	45.6	380	0%	0%
3	2	52.5	518	1	45.6	380	14%	27%
4	2	110	900	1	52.8	440	> 50%	> 50%
5	3	102	1445	1	56	800	45%	45%
6	2	77.2	1192	1	61.6	880	20%	27%
7	2	108.3	1675.7	1	71.4	1020	34%	40%

decisions early in the design phase, and to facilitate the use of SBCE-enabling tools (trade-off curves, limit curves, checklists and so on) for the exploration, communication and convergence of sets to avoid unnecessary design reworks and missing customer goals.

Discussion

The SBCE game has been played by more than 60 designers and managers working in a company that designs and develops innovative humidification and air-conditioning products. The game has been very effective in bringing hands-on experiences to practitioners about SBCE. According to the players, the effectiveness of the game for educational purposes can be summarised as follows.

First, the game helps them in having visibility in the early phase of design. In Stage 1, most teams follow trial-and-error approaches (PBCE) to find their solutions. In this stage, players lack visibility in terms of available alternative subsystems solutions, compatibility of subsystem solutions, feasibility of solutions and, finally, which solution was best for cost and time. This stage simulates the prevalent practices in the industry. Designers commit to a solution in an early stage of the PD process but later are subject to unnecessary reworks because of failures. As a result, success is not assured without iterating multiple times. For example, Table 5 shows the performances of different teams who played in the game session. When one looks at the number of trials, teams go through more iterations in Stage 1 than in Stage 2. This allows players to realise that following SBCE gives them the advantage of having more visibility early. Players can explore solutions and eliminate incompatible and infeasible ones. As a result, teams' performances in Stage 2 have improved, because fewer iterations and lower penalty costs have occurred. Moreover, SBCE helps in having the right number of components to add in the airplane – reducing the cost of components and the ordering time. The results shown in Table 5 might not reflect what happens in the real industrial implementation of SBCE; however, it does clearly teach about the benefits of SBCE in achieving visibility and improving performances in the early phase of the design process.

The second benefit of the game is related to helping players understand how the SBCE enablers work so that the players can make decisions based on knowledge. The enablers of SBCE, such as trade-off curves, limit curves and checklists, are systematically integrated in Stage 2 of the game. These enablers are needed to have a true convergence process in Stage 2 of the game. Practitioners acknowledge that these enablers are of paramount importance in capturing knowledge effectively for future reuse. All the teams who played used the same enablers but for different customer requirements. The enablers are used to generalise knowledge and to help standardise representation and capturing of design knowledge. However, the players underscore the fact that in real design practices, developing these enablers is challenging. In reality, considerable time and investment are needed to develop these tools to execute SBCE and to leverage knowledge reuse. This is a significant challenge they have to face to have a true SBCE process.

Conclusion

The SBCE process has been a subject for academic discussion for more than a decade. Industries have increasing interest in knowing and adopting SBCE in PD. To introduce and bring hands-on experiences to practitioners, the SBCE game shows significant potential. Before the actual implementation process, playing the game can create awareness among development teams. Furthermore, the benefits and challenges of adopting the SBCE process in practice can be discussed among team members during the game. Therefore, the game has significant educational benefits for industries.

The research does have its limitations, however. More research is required regarding the following aspects:

- *SBCE game design*. Future research might focus on including other SBCE enablers. The current version mediates only common tools and methods (trade-off curves, limit curves and checklists) as enablers of SBCE. However, several other enablers are crucial. For instance, those enablers related to managerial and organisational issues (e.g. the impacts of matrix organisation, the roles of the chief engineering system, the early-involvement suppliers, mentoring and coaching) can be added to the game. Moreover, an automated version of the game could enable its dissemination to wide audiences in industries and universities.
- *Validation of the game effectiveness*. To measure the game's effectiveness in teaching the principles of SBCE, research might focus on preparing measurement plans (e.g. questionnaires). Moreover, introducing the game to different industrial participants across different industries will be useful in future research to obtain detailed feedback from practitioners.
- *Drawing implications for practices*. In the extant SBCE literature, there is little evidence presented showing SBCE benefits (what are the effects of applying SBCE to improve performance?). Moreover, no implementation roadmaps exist to show stepwise guides to SBCE adoption in companies. Regarding these gaps, the game can be used to draw implications from practices. First, it can be used in future research to collect opinions on the benefits and challenges in adopting SBCE. Second, practitioners can systematically be interviewed or surveyed to set forth the steps in the implementation of SBCE in real projects. Therefore, the SBCE game not only helps educate, it can also be used to persuade companies to adopt it. If practitioners perceive its benefits and map implementation guides, the adoption of SBCE will be accelerated. For these purposes, the SBCE game is a suitable tool to facilitate future research.

Acknowledgement

The authors acknowledge the European Commission for its financial support and the project's partners for their comments and suggestions during the preparation of this article. Moreover, the authors wish to thank managers and designers at Carel Industries (www.carel.com) who have participated in the game. Special thanks to Mr Armin Akaberi for his valuable support and participation in the game development.

Funding

This research has been conducted as part of a European project titled 'Lean Product and Process Development' (LeanPPD, http://www.leanppd.eu) (NMP-2008–214090).

Declaration of conflicting interests

The authors declare that there is no conflict of interest.

Note

1. Min and Max boundaries are not considered. For example, for number of passengers, $Np = (91, 110)$, the customer accepts an airplane that carries a minimum of 92 and a maximum of 109 passengers.

References

Al-Ashaab A, Golob M, Attia UM, et al. (2013) The transformation of product development process into lean environment using set-based concurrent engineering – a case study from an aerospace industry. *Concurrent Engineering: Research and Applications* 21(4): 268–285.

Al-Ashaab A, Shehab E, Alam R, et al. (2010) The conceptual LeanPPD model. In: *Proceedings of the 17th ISPE international conference on concurrent engineering*, Cracow, 8–12 September 2010. London: Springer-Verlag.

Bernstein JI (1998) *Design methods in the aerospace industry: looking for evidence of set-based practices.* Master Thesis, Massachusetts Institute of Technology, Cambridge, MA. Available at: http://lean.mit.edu/component/docman/cat_-view/99-presentations/83-lai-annual-conferences/210-1998-plenary-conference?Itemid = 1193

Charsky D (2010) From edutainment to serious game: a change in the use of game characteristics. *Games and Culture* 5: 177–198.

Duke RD and Geurts JLA (2004) *Policy Games for Strategic Management: Pathways into the Unknown*. 1st ed. Amsterdam: Dutch University Press.

Egenfeldt NS (2006) Overview of research on the educational use of video game. *Digital kompetanse* 1: 184–213.

Geurts JLA and Joldersma C (2001) Methodology for participatory policy analysis. *European Journal of Operational Research* 128 (2): 300–310.

Cousens A, Goffin K, Mitchell R, van der Hoven C, Szwejczewski, M (2009) Teaching New Product Development Using the 'CityCar' Simulation. *Creativity and Innovation Management* 18(3): 176–189.

Hoheisel J, Weber F and Windhoff G (2000) New approaches for training and education of engineers by using simulation games. In: *IFIP TC5/WG5.7. Proceedings of the 4th international workshop on integrated production management systems and the European group of university teachers for industrial management*, Ghent, 26–29 November 1998, pp. 99–110. New York:Springer-Verlag US.

Hoppmann J (2009) *The lean innovation roadmap: a systematic approach to introduce lean in product development process and establishing a learning organisation*. Master Thesis, Technical University of Braunschweig, Braunschweig. Available at: http://lean.mit.edu/downloads/cat_view/75-theses/492-2009-theses

Inoue M, Nahm Y, Tanaka K, et al. (2013) Collaborative engineering among designers with different preferences: application of the preference set-based design to the design problem of an automotive front-side frame. *Concurrent Engineering: Research and Applications* 21(4): 252–267.

Kennedy MN and Harmon K (2008) *Ready, Set, Dominate: Implement Toyota's Set-Based Learning for Developing Products and Nobody Can Catch You*. Richmond, VA: Oaklea Press.

Khan M, Al-Ashaab A, Shehab E, et al. (2011) Towards lean product and process development. *International Journal of Computer Integrated Manufacturing*. Available at: http://www.tandfonline.com/doi/abs/10.1080/0951192X.2011.608723?tab = permissions#tabModule (accessed January 2013).

Liker JK, Sobek DK, Ward AC, et al. (1996) Involving suppliers in product development in the United States and Japan: evidence for set-based concurrent engineering. *IEEE Transactions on Engineering Management* 43(2): 165–178.

Mayer IS (2009) The gaming of policy and the politics of gaming: a review. *Simulation & Gaming* 40(6): 825–862.

Morgan JM and Liker JK (2006) *The Toyota Product Development System: Integrating People, Process, and Technology*. New York: Productivity Press.

Nahm YE, Ishikawa H and Yang YS (2007) A flexible and robust approach for preliminary engineering design based on designer's preference. *Concurrent Engineering: Research and Applications* 15(1): 53–62.

Oosterwal DP (2010) *The Lean Machine: How Harley-Davidson Drove Top-Line Growth and Profitability with Revolutionary Lean Product Development*. New York: American Management Association.

Pourabdollahian B, Taisch M and Kerga E (2012) Serious games in manufacturing education: evaluation of learners' engagement. In: *VS-GAMES 2012. Proceedings of the 4th international conference on games and virtual worlds for serious application*, Genoa, 29–31 October 2012, vol. 15, pp. 256–265. Amsterdam: Elsevier.

Prensky M (2001) *Digital Game-Based Learning*. e-book ed. New York: McGraw-Hill.

Raudberget D (2010) Practical applications of set-based concurrent engineering in industry. *Journal of Mechanical Engineering* 56(11): 685–695.

Riedel JCKH and Pawar KS (2001) Academic and industrial user needs of a concurrent engineering computer simulation game. *Concurrent Engineering: Research and Applications* 9(3): 223–237.

Rossi M, Kerga E, Taisch M, et al. (2012) Learning methodologies to diffuse lean product development to industries. In: *PLM 2012. Proceedings of international conference on product lifecycle management, IFIP*, Montreal, QC, Canada, 9–11 July 2012, pp. 287–298. Berlin Heidelberg:Springer.

Sobek DK (1997) *Principles that shape product development systems: a Toyota and Chrysler comparison*. Unpublished PhD Thesis, Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI.

Sobek DK, Ward AC and Liker JK (1999) Toyota's principles of set-based concurrent engineering. *Sloan Management Review* 40: 67–83.

Takai S (2010) A lifecycle-cost model for set-based concept testing. *Concurrent Engineering: Research and Applications* 18(2): 121–127.

Ward AC, Liker JK, Cristiano JJ, et al. (1995) The second Toyota paradox: how delaying decisions can make better cars faster. *Sloan Management Review* 36: 43–61.

Ward AC, Shook JK and Sobek DK (2007) *Lean Product and Process Development*. Cambridge, MA: Lean Enterprise Institute.

Wouters P, van der Spek ED and Oostendorp H (2011) Measuring learning in serious games: a case study with structural assessment. *Educational Technology Research and Development* 59(6): 741–763.

Zhang X, Huang H, Wang Z, et al. (2013) A Pareto set coordination method for analytical target cascading. *Concurrent Engineering: Research and Applications* 21(4): 286–295.

Zyda MJ (2005) From visual simulation to virtual reality to games. *IEEE Computer Society* 38(9): 25–32.