SIMULATION OF FEEDBACK LOOPS IN ENGINEERING DESIGN

Tapia, Francisco (1);
McKay, Alison (2);
Robinson, Mark (3)

1: University of Leeds;
2: University of Leeds;
3: University of Leeds

ABSTRACT

Feedback loops are a key characteristic of engineering design processes that increase complexity, time to market, and costs. However, some feedback loops, due to design iteration, have a positive impact on design outcomes (i.e., the quality of the final design), so are worth the time and costs incurred. Other loops, resulting from rework, also have a positive impact on the final design but their impact on current projects, in terms of their urgency and so interruption, is high. Thus, overall, and drawing on socio-technical systems literature, some feedback loops are virtuous circles with a positive impact whereas others are vicious circles with a negative impact. In this paper, we report early work exploring these interplays between rework and design iteration through the development of process simulation models.

Keywords: Simulation, Process modelling, Decision making

Contact:
Tapia, Francisco
University of Leeds
iDRO, Institute of Design, Robotics and Optimisation.
United Kingdom
mnftl@leeds.ac.uk

Cite this article: Tapia, F., McKay, A., Robinson, M. (2021) 'Simulation of Feedback Loops in Engineering Design', in *Proceedings of the International Conference on Engineering Design (ICED21),* Gothenburg, Sweden, 16-20 August 2021. DOI:10.1017/pds.2021.527

1 INTRODUCTION

Product development processes codify ways in which manufacturing organisations deliver new products to market in response to customer demands and strategic priorities. These processes involve a series of stage-gates, where decisions to proceed or not drive such projects and where, given the requisite quality, key performance indicators are cost and time to market. This paper focusses on time to market, which, in ideal circumstances, would be as fast as possible and so each gate would only be passed through once. However, there are inherent trade-offs because completing a stage too quickly risks compromising quality, which can lead to rework if only uncovered at a later stage in the project. Rework is a negative feature because it consumes time (Arundarachawat et al., 2009) and therefore affects project duration and cost. According to Smith and Eppinger (1997), rework arises from changes in requirements or task repetition due to initially imperfect information.

In contrast, engineering design processes are technical processes through which innovations are developed and embedded in products. They provide frameworks for stages of product development projects that include the development of designs for the whole product and its parts, which themselves may be regarded as independent products. Typically, it is the creativity and capability of the engineering design teams that govern their performance in engineering design processes. Engineering designers prioritise technical quality and fulfilling all design requirements, but their activities are carried out in the context of time pressures. In engineering design, iteration improves quality by systematic exploration and understanding of the complexity of design problems, leading to more efficient solution-finding processes (Le, Wynn, and Clarkson, 2010). Design iteration enables the progressive generation of knowledge, concurrency, and integration of necessary changes (Wynn and Eckert, 2017). Although design iteration adds time to design activity and product development, it has the potential to improve design quality and so reduce rework in product development projects. Hence, a key challenge for design managers lies in establishing a balance between positive feedback loops, in the form of design iteration, and negative feedback loops, in the form of avoidable rework.

This paper explores interplays between these different kinds of feedback loops with a view to producing tools that design managers can use to inform decisions about allocating resources and when to stop design iteration to complete design tasks on time. In Section 2, three core areas of literature are reviewed: in Section 2.1, socio-technical systems approaches; in Section 2.2, feedback loops found in engineering design processes, including rework and iteration; and finally, in Section 2.3, the simulation of engineering design processes. Sections 3 and 4 provide overviews of the research approach and design case study used. Section 5 reports the results of discrete event simulations of the case study's stage-gated design process, focussed on rework. Then, Sections 6 and 7 use these results to inform design requirements for agent-based simulations of design activities that include design iteration.

2 BACKGROUND

Engineering design can be seen as a socio-technical system because both the stage-gated processes within which design teams work and the activities of individual designers are systems involving people, and so human and organisational behaviours, and technology. For this reason, in Section 2.1, we review literature on socio-technical systems thinking and use it to frame the social and organizational aspects of the design process. This is followed, in Sections 2.2 and 2.3, by a review of literature on design iteration and rework, and previous work simulating design processes.

2.1 Socio-Technical Systems approaches

Socio-technical systems are developed to perform specific tasks. They include technical aspects such as technology, infrastructure, and processes; socio aspects, such as people, goals, and culture; and the systemic connections between these (Clegg et al., 2017). The outcome of the analysis of these systems is a better understanding of how human and organisational factors influence task performance and how those technical systems are used.

The new product development processes within engineering design domains are representative examples of complex socio-technical systems, as design is a social process involving team-working, complex problem solving, creativity, and information exchange (Robinson 2016). During these processes, designers communicate their ideas by different means and record them in documents as their understanding of the design solution evolves. Changes in designers' ideas potentially add

https://doi.org/10.1017/pds.2021.527 Published online by Cambridge University Press

uncertainty, requiring more exploration of alternatives and making the design process iterative (Piccolo et al. 2018).

2.2 Vicious and virtuous circles and iteration

Feedback (or causal) loops are an important concept in social and organisational theory, and enhance understanding of the relationships between the past and current state of a system (Tsoukas and e Cunha 2017). A causal loop that tends to reinforce or amplify a change is called a positive, reinforcing, or deviation-amplifying feedback loop, while a closed loop that tends to counteract a change is called a negative, deviation-counteracting feedback loop (Masuch 1985). A vicious circle is a deviation-amplifying loop that worsens a challenging situation, while a virtuous circle is a deviation-amplifying loop that improves a good situation (Tsoukas and e Cunha 2017). Vicious and virtuous circles are particularly common in social systems, such as organisations, where there are numerous heterogeneous and often conflicting causal loops.

Within product development, both strategy and early design decisions influence the organisational structures needed to develop engineering designs and so the social networks formed by design teams. As designs develop, new information and constraints emerge and design requirements change, leading designers to revisit and revaluate design decisions. These iterations contribute to the quality of the design and progression through development processes by facilitating knowledge generation (Wynn and Eckert 2017). However, iterations also increase project duration and cost, and can cause rework when these iterative cycles propagate into different stages. Love (2002) defines rework as the unnecessary effort and delays arising from redoing a process or activity not adequately implemented the first time.

Cho and Eppinger (2001) argue that rework occurs for three reasons: (1) new information obtained from overlapped tasks after work has started; (2) inputs changing because of rework on other tasks; and (3) outputs failing to meet established criteria. They also distinguish between feedback and feed-forward rework. Feedback rework occurs because a downstream task fails to meet established criteria, and feed forward rework occurs on a downstream task because new information arises from an upstream task. In this paper, we simulate how these two kinds of feedback loop (rework and iteration) lead to vicious and virtuous circles in product development systems.

2.3 Computer simulation of engineering design

A number of frameworks and modelling methods have been proposed to improve the performance of business systems. These approaches use emerging computer simulation tools to enhance the design, management, and evaluation of such systems. Mykoniatis and Angelopoulou (2020) provide a framework for integrating different simulation methods in different domains such as socio-technical systems, cyber-physical systems, business and healthcare organizations. Agent-based simulations of new product development processes seek to integrate in a much more sophisticated way the social interaction between agents, including skill levels and behaviours (see Hassannezhad et al., 2019; Fernandes et al., 2017; Crowder et al., 2012). Other simulation methods are now integrating hybrid simulation and multi-paradigm simulation to capture the complexity of real systems, providing a more comprehensive and holistic view of the system under investigation.

Djanatliev and German's (2013) framework for multi-paradigm simulation models in healthcare captures interactions between different abstraction levels in real-world socio-technical systems. At a macro level, there are fewer details and a high abstraction level is required, so this is more appropriate for a holistic view including multiple product development projects and is therefore unsuitable here. However, of high relevance here are meso-level models, which cover tactical level interactions in medium detail, such as product development, and micro-level models which cover operational level interactions in fine detail, such as engineering design activities. From an engineering design perspective, three approaches to modelling and simulation of iterative process have been identified: (1) task-based approaches, where the iteration is a repetition of the task already completed or similar tasks in different contexts; (2) actor-based approaches, where iteration involves a continuous dialogue between agents to coordinate; and (3) information-based approaches, in which process information determines process behaviour (Wynn, Eckert, and Clarkson, 2007).

https://doi.org/10.1017/pds.2021.527 Published online by Cambridge University Press

3 APPROACH

A key feature of the processes simulated here are two kinds of feedback loop: rework that is governed by stage-gated processes (and therefore well-suited to discrete event simulation) and design iteration that is driven by individual designers (and so best modelled using agent-based simulation). As a result, a multi-paradigm simulation approach was needed. We used that proposed by Mykoniatis and Angelopoulou (2020) (integrating agent-based, discrete event, and systems dynamics simulation approaches) which includes four phases: (1) conceptual modelling, (2) simulation model development, (3) verification and validation, and (4) results and documentation. In addition, they identify three important questions when using their framework: (1) Why and when does a real-world system require multi-paradigm modelling and simulation? (2) What are the interaction points among the different simulation models used? (3) How do the simulation models interact with each other to exchange information?

The first question was answered earlier in this section, when we identified key characteristics of design iteration and rework that require a multi-paradigm approach. The results reported in this paper relate to the discrete event simulation of rework processes. In part, the work presented was directed towards answering the second and third questions, as we discuss in Sections 6 and 7. To this end, the discrete event model development has been carried out with cognisance of the requirement to integrate with agent-based models of design iteration. For this reason, the conceptual models reported later in the paper take account of and begin to answer the last two questions.

Within Mykoniatis and Angelopoulou's first stage (conceptual modelling) we used Nikolic and Ghorbani's (2011) methodological approach for the development of simulations of complex sociotechnical systems, as shown in Figure 1. In this research, the first two stages map onto Mykoniatis and Angelopoulou's (2020) Phase 1, and the final three stages map onto their Phases 2-4. In the remainder of this section, we describe the approach used before introducing the case study and results.

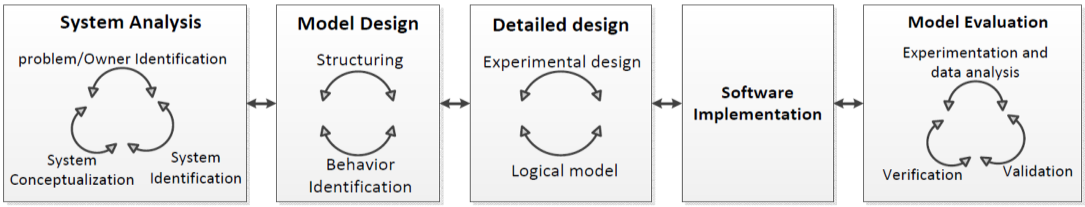

Figure 1: Methodological framework. Reproduced from Nikolic and Ghorbani (2011).

3.1 Conceptual Model Development (Phase 1)

Nikolic and Ghorbani (2011) identify the need for a systematic approach to model development as part of an ongoing process for standardising modelling practice. These stages are common in software engineering design methodologies but include several iterative sub-steps specific to simulation modelling. Their first two stages, System Analysis and Model Design, fall into Phase 1 of this research and include establishing the purpose of the models and identifying the problem being simulated, key stakeholders, and the system to be conceptualised. Next, in model design, for agent-based modelling, agents and interactions between them are identified. In this research, the model design process was applied to the discrete event simulation models where key process stages and events were identified. However, given the need for a multi-paradigm approach, the development of the discrete event models also identified key actors and their behaviours with a view to defining requirements for future agent-based models that represent human behaviours and design iteration. In addition, building the initial models in this way helps answer Mykoniatis and Angelopoulou's (2020) second question related to interaction points between the models. In this paper, the key process stages used are reported in Section 5.1. However, a challenge when modelling design processes lies in the fact that the structure and steps in the process to be simulated depend in part of the product being designed and decisions made in the design process itself. For this reason, the process reported in Section 5.1 was applied to the case study introduced in Section 4.

3.2 Model development, verification, validation, and reporting of results (Phases 2-4)

The final three phases of the process used in this research align with Stages C (Detailed Model Design), D (Software Implementation), and E (Model Evaluation) of Nikolic and Ghorbani's process. The focus

https://doi.org/10.1017/pds.2021.527 Published online by Cambridge University Press

of this paper lies in the design of the models and results generated to help answer Mykoniatis and Angelopoulou's final two questions. For this reason, we introduce the simulation model that was implemented in Section 5.2 and, in Section 0, the results generated and the requirements they create for the next stage of the research, namely, the design of future agent-based simulation models.

4 CASE STUDY

A generic new product development process provides a sequential progression of development tasks (Artmann 2009). Its typical representation is as a stage-gate model, which includes stages with predefined checkpoints (gates) that contain deliverables for each functional area that must be passed to proceed. This stage-gate model provides a chronological structure (the process) which, for the design of a given product, is combined with a product architecture to form the development process structure. In this paper, the design of a bicycle handlebar assembly (Figure 2) is used as a case study. The system decomposition of the handlebar assembly identifies the teams needed to design the subsystems: the brake lever design team, the gear change design team, the handlebar design team, and the handlebar assembly integration team. The simulations reported in Section 5 focus on the three design teams.
The bicycle development process starts when a design request is delivered simultaneously to the brake lever and gear change design teams by the handlebar assembly team. The handlebar assembly integration team, however, must wait for the two designs to conclude its process. Each design team iterates the design for each component. However, in some cases, these feedback loops are coordinated (i.e., communicated effectively and on time), while in other cases, they are not (i.e., not communicated or communicated with a delay). Where not coordinated, iterations lead to rework, and if the rework is not coordinated then that might lead to further rework, in a vicious circle.
The simulation model introduced in the next section captures interactions between these different teams and processes to explore the influence of iteration and rework on system performance, considering the possible increment on the workload and the introduction of delays, but also improvements in the quality and efficiency of the teams.

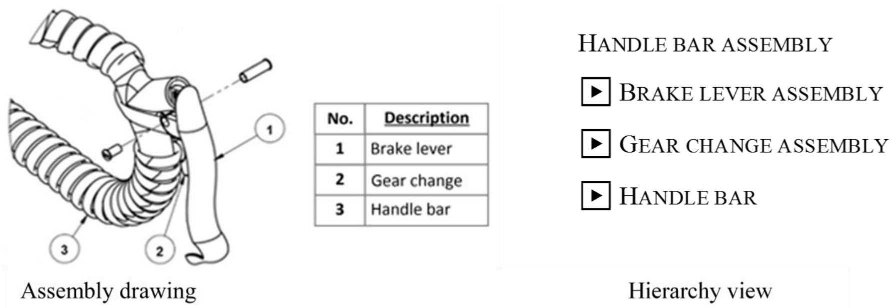

Figure 2: Handlebar assembly case study

5 RESULTS

In this section we present the results from applying the research approach (Section 3) to the case study (Section 4).

5.1 Conceptual Model Development (Phase 1)

Using Djanatliev and German's (2013) framework for multi-paradigm simulations in complex socio-technical systems, different abstraction levels (micro, meso, and macro) and relationships between them were identified in the case study. This resulted in the identification of actors, relationships, behaviours, and possible states (see Figure 3 and Table 1), and provided necessary elements for the development of a conceptual model to underpin the simulation of the case study design process (see Figure 4 and Table 2). In this paper, we report on the development of a discrete event simulation model that captures aspects of the design process governed by the overarching product development process within which it exists.

5.1.1 System analysis

Product development processes are multidimensional by nature, comprising product architecture, communication patterns, iterations and rework as key features that interact (Yassine, 2018) within the three domains of task quality, project schedule, and design teams.

https://doi.org/10.1017/pds.2021.527 Published online by Cambridge University Press

5.1.2 Systems identification

The system decomposition takes as its staring point Djanatliev and German's (2013) framework diagrams, as shown in Figure 3. In our case study, at the micro-level there are the design teams and the designers, their characteristics and behaviours. At the meso-level, there are the process, the progression of the tasks, and the process project schedule domain. Finally, at the macro-level, there are the upstream project performance and quality of the tasks, assessment of rework, iteration, and task approval.

5.1.3 System conceptualization

This stage formalizes concepts, relationships, behaviours, and interactions, with the objective of removing ambiguities in preparation for an interpretable computational language. It also captures the actors' relationships, interactions, actions, behaviours, and states. The outcome from this stage is Table 1, where actor interactions, behaviours, and states, were identified.

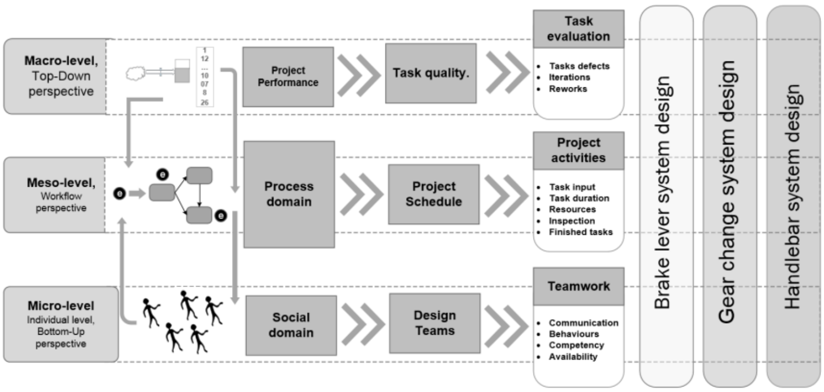

Figure 3: System identification adapted form Djanatliev and German (2013)

Table 1: System conceptualization of the handlebar system design for a bicycle

Actors	Relationship	Interaction	Actions	Behaviours	States
Brake lever system design	Sub-systems for the handlebar assembly system	Affects the handlebar design	Perform design task	Works in the design task	Performing design task
		Affects the gear change design		Determines the need for information	Asking for information
Gear change system design		Is affected by the brake lever and gear change design	Exchange information	Decides to carry on	Answering questions
		Is affected by the iteration of the tasks		Rejects the information received	Waiting for information
Handlebar system design			Evaluate information	Evaluates the information received	
		Provides design information		Waits for information from other designer	Iterating

5.2 Simulation model design

The conceptual model derived from the systems identification and conceptualization is represented in the UML activity diagram in Figure 4, also showing the interactions between three teams. The narrative developed for the conceptual model is as follows: (1) The design cycle starts with the reception of one or more design requests. (2) Design teams receive the information simultaneously. (3) Design teams for the Brake lever and Gear change start working immediately. (4) However, the Handlebar design team must to wait for the Brake lever and Gear change designs to be able to carry on with its design task. (5) Each team develops its designs in an allocated amount of time. (6) At the decision gate, it is decided which designs are sent for feedback, and so design iteration, and which continue to the next phase.

5.2.1 Detailed design

Within this framework, the relevant activity in this stage is to make sure that identified concepts can be implemented in a computational language while retaining their original meaning. In Table 2, below, the main concepts have been translated into the simulation modelling blocks and initial parameters are established. The discrete events simulation diagram adapted with arrows depicts the model design

https://doi.org/10.1017/pds.2021.527 Published online by Cambridge University Press

implemented in the AnyLogic 8 package (https://www.anylogic.com/#tab1). This package supports the development of hybrid simulation models that combine discrete event, system dynamics, and agent-based approaches. This paper reports results from a discrete event simulation model that is part of a wider, hybrid, modelling exercise.

5.2.2 Simulation results

The discrete event simulation model presented in this paper is an early result of an integrative analysis that is considering how design iteration, rework, and the social interactions within and across design teams influence the progression and quality of design tasks in the new product development systems. The application of the model is currently limited to changes in the inputs, and their impact on time taken, i.e., the time taken by one "design request" entity to pass through the simulation system until it arrives at the "approved design" block. Design requirements are fed into the model and transformed into designs. The number of designs produced is influenced by the number of feedback loops that occur. The model includes two kinds of feedback loop. Iterations are randomly selected feedback loops that occur within the design of a given part whereas rework loops span the design processes of multiple parts and are modelled as service delays to queues. Together, these influence the time taken to produce a design and so, in a simulation experiment with a fixed runtime, the number of designs produced.

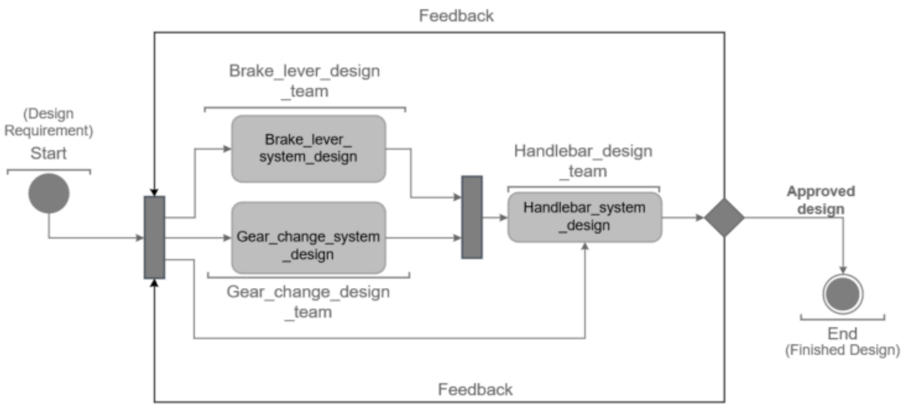

Figure 4: UML activity diagram for the discrete events conceptual model.

Table 2: Table of initial parameters for discrete events simulation

	Component	Name	Justification	Qty	Icon	Parameter/ Variable	Parameter set
1	**Entities** The design task requirement	**Input/source**	Generates the entities and adds them to the process	4		Rate, per moth	1,,,4 requirements per month
	From a previous stage injects tasks to design teams inputs/ at the end receives the entities leaving the model	**End/Sink**	Disposes of the entities leaving the model	2		N/a	On enter: (Input. Inject (1); Input1.inject (1); Input2.inject (1);)
2	**Activities** Perform design task	**Service**	The block is a combination of three inner blocks: Seize, Delay, and Release that simplifies modelling common process, simulates design task service.	2		Delay time/ Queue: exit on timeout	Delay Time: uniform(4, 7); Exit in timeout: uniform(16,48)
3	Perform integration design task		Assembles entities (from one to five sources) into a single entity based on a predefined bill of material.	1		Delay time	Delay Time: uniform(4, 7.5)
4	**Routing** Routes the elements(documents, designs) back to the design activities	**SelectOutputIn/ SelectOutputOut**	To implement a custom routing to N flowchart branches,	4,/1		Probability/ Percent	Percent: uniform(0.7, 0.9)
5	**Queue** Used to avoid conflicts in the return of the elements (documents, designs)back to the design task	**Queue**	Stores entities that cannot move forward in the process immediately	7		N/a	None
6	**Resources** Assigned resources to the activities	**Resoursepool**	Defines and stores the resource unit(s) of a certain type.	3		Capacity defined by schedule	Schedule: Monday, Wednesday and Friday form 9am to 16pm; Units: 4

In these simulations, the run-time is set to 156 weeks, where one simulation system tick represents one week and a month is 4.3 ticks.
The input parameter to the simulation model is the number of design requirements, which has a direct impact in the project performance affecting both time in the system and the number of finished designs

https://doi.org/10.1017/pds.2021.527 Published online by Cambridge University Press

counted in the "approved design" block at the end of the runtime. The output of the simulation model is the number of designs produced in the runtime of the simulation experiment; this depends on the number of iterations and volume of rework given a fixed capacity.
To analyse the impact of changes in inputs in the form of design requirements, an experiment was run starting with one, two, three, and four requirements per month, running 40 times on each configuration.
The time plot represented in Figure 6(a) depicts iterations data generated by each configuration with 156 weeks model run time in the "x" axis. The plots representing one, two, and three requirements per month reflect a progressive incremental behaviour in the accumulated number of iterations the "y" axis. However, the plot line for the four requirement inputs depicts only a small difference among the third input experiment in the analysed data.
In Figure 6(b), the time plot chart shows data for the reworks recorded during the four input experiments. The analysis of data shows that with one and two requirement inputs, rework rate is not higher than 0.34 with respect to iteration, but in the three and four requirements input, the rework rate is 1.89 and 5.34 respectively. The data generated with this partial implementation suggest that a lower iteration rate does not necessarily mean a lower rework rate, as rework may have different sources. In the current state of the partial implementation of the model, rework data in experiments three and four are not yet reflecting yet a real world system. Further data collection and a gradual release of segments of the model should be performed in order to completely validate the model.

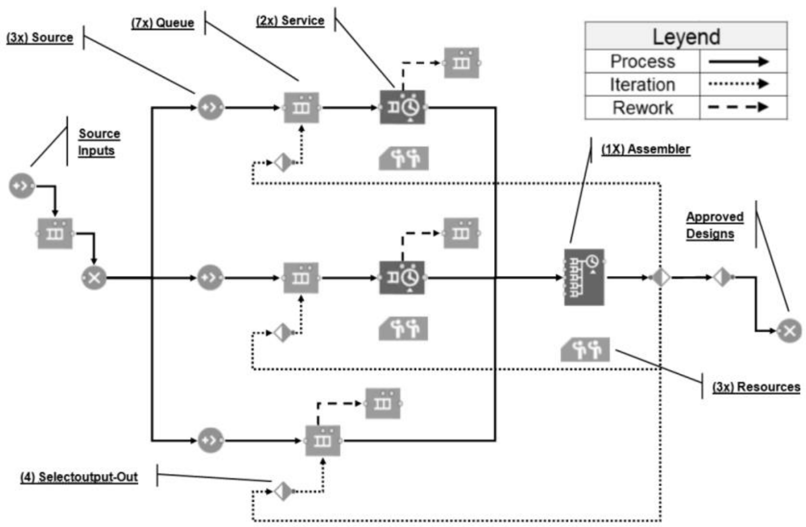

Figure 5: Discrete events simulation diagram implemented in AnyLogic 8.

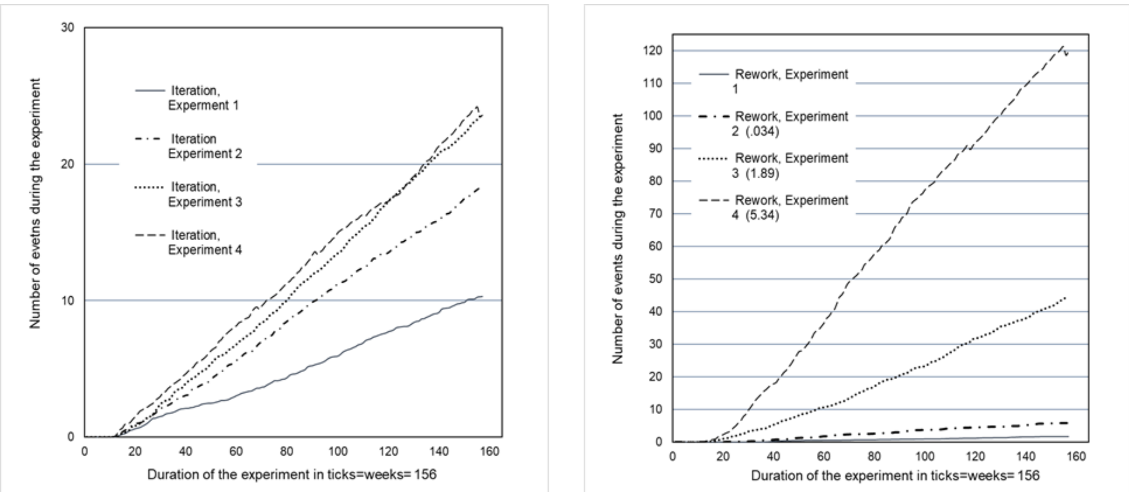

Figure 6: Time plots showing results for (a) iterations and (b) rework for each experiment.

https://doi.org/10.1017/pds.2021.527 Published online by Cambridge University Press

6 DISCUSSION

The simulation model introduced in this paper reflects the complexities of real-world design processes by combining a general purpose design process framework and the architecture of the product being designed. With such a model, we are able to delineate specific feedback loops that occur in the process and so produce more realistic simulations of the design process than those that do not take account of the product being designed. Although not yet reflected in the model, the product architecture evolves through the process and so could be used to inform simulations of subsequent stages of the design process which, in turn, could be used to inform design decisions. For example, if alternative product architectures were being considered then the kinds of simulation models introduced here could be used to estimate the design time needed for each option. The results presented are preliminary, and have not yet been validated, but are presented here to illustrate the potential of the overall approach. The feedback loops captured in the simulation model span process stages.

From literature on vicious and virtuous circles and design iteration, we recognise that, in addition to these feedback loops, which can usefully be seen as a form of rework, there are feedback loops that reflect the iterations of individual designers and design teams. These occur in the context of stage-gated processes, e.g., as shown in Figure 7. The development of the discrete event simulation model, where the feedback loops represent rework, has allowed us to formulate requirements for the next stage of the work reported in this paper which is to incorporate agent-based models that represent design iteration as micro-level feedback loops. Building on Mykoniatis and Angelopoulou (2020), the main discrete events outputs have been identified, establishing the temporal aspects of the tasks and the conceptualization of the system. The simulation is able to report the typical outputs of a discrete events simulation: time in the system, time in a queue, time to be served, and resources utilization. In the current simulation models, rework and iteration are characterized as random events that occur during the execution of the task. For this reason, they are quantified as numbers of events in the simulation runtime, but with the implementation of the agent-based model, iteration and rework will be triggered by states and conditions happening at the micro-level, with the agents (designers) sharing and requesting information, and influencing the design task at the meso-level of the discrete events simulation.

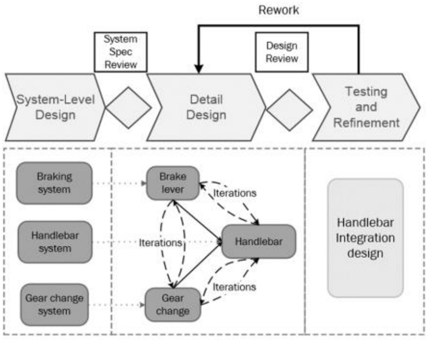

Figure 7: Design iteration within the design process

7 CONCLUSIONS AND FURTHER WORK

This paper reports results of early work on the development of simulation models of feedback loops in design processes. We used a framework that supports the integration of three kinds of simulation model: discrete event, agent-based, and systems dynamics. Our focus on the discrete event simulations has enabled the identification of requirements for associated agent-based models by providing insights on Mykoniatis and Angelopoulou's (2020) second and third questions that surround the location of interaction points among the different simulation models and the ways in which these models interact with each other to exchange information. A key interaction point lies between the part design activities in the discrete event model and the agent-based models that capture designers' behaviours. In the current model, design events are allocated a length of time in which the design process is completed and design iterations are determined at stage gates and so span design steps for different parts. By integrating this model with agent-based models, we will also be able to explore design iterations within the designing of individual parts.

In the longer term, future systems dynamics models will contribute to the study of non-linear behaviours at a product development system level. However, to add value in an engineering design context, they require integration with detailed models such as those introduced in this paper. A key feature of these models, and a challenge in simulating design processes, lies in the close relationship between the structures of design process and the product architecture that is developed through the design process itself. This paper paves the way for the establishment of digital twins of design processes, and so information communicated through both feedback and forward loops in design processes, carried out within single organisations and across supply networks. In turn, such models could be used to identify optimal process pathways for a given design process and so inform the design of the product itself.

ACKNOWLEDGMENTS

The authors thank the Consejo Nacional de Ciencia y Tecnología of Mexico (CONACYT) for the financial support provided to Francisco Tapia Lara as part of his PhD Scholarship (472443/843527) at the University of Leeds.

REFERENCES

Artmann, C. 2009. The value of information updating in new product development (Vol. 620). Springer Science & Business Media.

Arundachawat, P., Roy, R., Al-Ashaab, A., Shehab, E. 2009. Design rework prediction in concurrent design environment: current trends and future research directions. In Proceedings of the 19th CIRP Design Conference–Competitive Design. Cranfield University Press.

Cho, Soo-Haeng, and Steven D. Eppinger. "Product development process modeling using advanced simulation." In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, vol. 80258, pp. 85-94. American Society of Mechanical Engineers, 2001.

Clegg, C.W., Robinson, M.A., Davis, M.C., Bolton, L., Pieniazek, R., & McKay, A. (2017). Applying organizational psychology as a design science: A method for predicting malfunctions in socio-technical systems (PreMiSTS). Design Science, 3, 1–31.

Crowder, R.M., Robinson, M.A., Hughes, H.P.N., Sim, Y.W. 2012. The development of an agent-based modeling framework for simulating engineering team work. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 42(6), pp.1425-1439.

Djanatliev, A. and German, R., 2013, December. Prospective healthcare decision-making by combined system dynamics, discrete-event and agent-based simulation. In 2013 Winter Simulations Conference (WSC) (pp. 270-281). IEEE.

Eppinger, S.D. and Browning, T.R., 2012. Design structure matrix methods and applications. MIT press.

Fernandes, J.V., Henriques, E., Silva, A., Pimentel, C. 2017. Modelling the dynamics of complex early design processes: an agent-based approach. Design Science, 3.

Hassenezhad, M., Cantamessa, M., Montagna, F., Clarkson, P.J. 2019. Managing Sociotechnical Complexity in Engineering Design Projects. Journal of Mechanical Design, 141(8).

Le, H.N., Wynn, D.C. and Clarkson, P.J., 2010. Evaluating the positive and negative impact of iteration in engineering processes. In Modelling and Management of Engineering Processes (pp. 89-100). Springer, London.

Love, P.E., Edwards, D.J. and Smith, J., 2016. Rework causation: Emergent theoretical insights and implications for research. Journal of Construction Engineering and Management, 142(6), p.04016010.

Masuch, M. 1985. Vicious Circles in Organizations. Administrative Science Quarterly, Vol. 30, No. 1 (Mar., 1985), pp. 14-33.

Nikolic, I. and Ghorbani, A., 2011, April. A method for developing agent-based models of socio-technical systems. In 2011 International Conference on Networking, Sensing and Control (pp. 44-49). IEEE.

Mykoniatis, K. and Angelopoulou, A., 2020. A modeling framework for the application of multi-paradigm simulation methods. Simulation, 96(1), pp.55-73.

Piccolo, S. A., Maier, A. M., Lehmann, S., McMahon, C.A. 2018. Iterations as the result of social and technical factors: empirical evidence from a large-scale design project. Research in Engineering Design, 30(2), pp. 251-270.

Robinson, M.A., 2016. Quantitative research principles and methods for human-focused research in engineering design. In Experimental design research (pp. 41-64). Springer, Cham.

Robinson, S. 2008. Conceptual modelling for simulation Part II: A framework for conceptual modelling. Journal of the Operational Research Society, 59(3), pp.291-304.

Smith RP, Eppinger S.D. 1997. Identifying controlling features of engineering design iteration. Management Science 43(3):276–293.

Tsoukas, H., e Cunha, M.P., 2017. On organizational circularity. The Oxford handbook of organizational paradox: Approaches to plurality, tensions, and contradictions, pp.393-412.

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

Wynn, D.C., Eckert, C.M. and Clarkson, P.J., 2007. Modelling iteration in engineering design. In DS 42: Proceedings of ICED 2007, the 16th International Conference on Engineering Design, Paris, France, 28.-31.07. 2007 (pp. 693-694).

Yassine, A., 2018, June. A Three-Dimensional View of Complex Product Development Management. In 2018 IEEE Technology and Engineering Management Conference (TEMSCON) (pp. 1-6). IEEE.