Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.

AN MVP APPROACH TO DEVELOPING COMPLEX HYBRID SIMULATION MODELS

William Jones
Philip Gun
Mehdi Foumani

Rio Tinto Centre for Mine Automation
University of Sydney
8 Little Queen Street, Chippendale
Sydney, NSW 2008, AUSTRALIA

ABSTRACT

Simulation is increasingly applied to capture highly complex systems and tackle ever more complex problems. We present a novel framework that aids modellers in undertaking long, highly complex simulation studies. Our framework provides a structured method guiding how a study will develop, aiding modellers to conceptualise a suitable model and plan how it will be coded. The approach incorporates the principle of minimum viable product development, a process directed toward satisfying the minimum requirements of the client as quickly as possible by deconstructing them into defined steps. The framework has wide applications, but, its modularity is particularly well suited to complex hybrid simulation projects which integrate multiple techniques into a single model. Our framework ensures value is generated for the stakeholders early in a project. It facilitates agile development involving stakeholders closely in the inevitably long project life-cycle, responding to their feedback as the project progresses to maximise value.

1 INTRODUCTION

The progression of a simulation project is often described with reference to a life-cycle loop, i.e., conceptual modelling (CM), model coding, experimentation, and implementation (Robinson 2004). This life-cycle can represent the progression of any project at a high level, however, it does not express detail of the process by which complex simulation models emerge over long projects. With the emergence of specialist software and the growth in available computing power over the past two decades, simulation has been applied to capture increasingly complex systems and tackle increasingly complex problems. Hence, highly complex simulation models are now common (Chwif et al. 2000). Notably, hybrid simulation (HS) has emerged as a popular approach for modelling complex systems (Brailsford et al. 2018), its popularity growth mirroring the increase in computing power. Highly complex models may be developed over periods of many months or even years. HSs may consist of distinct modules which can potentially be developed by separate teams of modellers in parallel. Numerous iterations of the conceptual model and model code may be developed before experimentation can begin with a full-system model. Throughout the long development time, stakeholder requirements and therefore modelling requirements may change.

We present a simulation life-cycle framework for the development of complex models. Our cross-disciplinary hybrid modelling framework, Type E - see Mustafee et al. (2020), rethinks the progression of a simulation study utilising techniques from outside the operations research literature. It breaks the development of a model into stages and incorporates the principles of Minimum Viable Product (MVP) (Ries 2011) and modular software development (Stannett and Dickinson 1990). For each stage, it illustrates how artefacts of the simulation model emerge (Robinson 2013). Completion of all development stages would see the emergence of a full-system simulation model, ready for full-system experimentation and the

978-1-6654-7661-4/22/$31.00 ©2022 IEEE

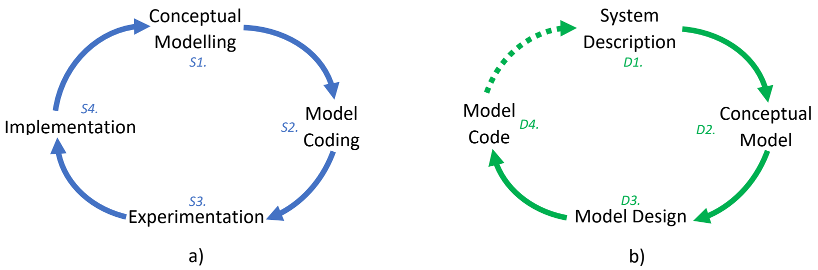

Figure 1: a) The simulation life-cycle, b) The artefacts of simulation model development.

completion of the life-cycle (Robinson 2004). Although the principles are applicable to any simulation study, the modular approach of our framework is particularly well suited to the development of complex HS projects which integrate multiple techniques into a single model.

The paper proceeds with a review of relevant background material (Section 2) including describing a successful completion of a typical simulation life-cycle loop (Robinson 2004). We then present our framework (Section 3), highlighting the similarities and differences with the process described in the previous section. Key to our framework is dividing the model development into stages. We provide guidance on how to do this (Section 4). We illustrate our frameworks with an example (Section 5) before discussing its benefits in terms of the model development process and project planning and management (Section 6). Finally, we give conclusions (Section 7).

2 BACKGROUND

Existing literature has described and categorised the various stages of a simulation study and several frameworks of differing levels of granularity have been proposed to illustrate the so-called *simulation life-cycle* (Kreutzer 1986; Pidd 1988; Robinson 2004). While the framework in Pidd (1988) consists of just three stages, the framework in Kreutzer (1986) is composed of nine. However, the process described is similar in each. Robinson (2004) defines four stages of the life-cycle (Figure 1a). We step through the four stages of Robinson's life-cycle (expressed in layman's terms), illustrating the roles of *clients* and *modellers* and the key features of each step, as follows.

S1. *Modellers* work with the *clients* to undertake CM activity.

Robinson (2013) defines a CM as "a non-software specific description of the computer simulation model (that will be, is or has been developed), describing the objectives, inputs, outputs, content, assumptions and simplifications" (p. 380). Key to the CM process is understanding the system to be modelled and, in many scenarios, vital to this is collaboration with stakeholders. The benefit of engaging stakeholders in the modelling process has been explored (Kotiadis et al. 2014; Robinson et al. 2014) and subsequently, guidance for modellers undertaking such collaborations has been developed (Tako and Kotiadis 2015).

CM is arguably the most important activity in the simulation life-cycle, however, it is poorly understood (Tako et al. 2019). HS has emerged as an approach for modelling complex systems, growing in popularity over the past two decades (Brailsford et al. 2018). Although the benefits of HS have been demonstrated across a range of applications (see examples discussed in Eldabi et al. (2016), Mustafee et al. (2017), Brailsford et al. (2018), Eldabi et al. (2019)), few studies reference a methodology applied during the CM activity. Brailsford et al. (2018) found CM to be one of the least explored areas in HS research.

Robinson (2020) identified the development of CM frameworks as a 'grand challenge' of the field. Brailsford et al. (2018) notes that concerning HS "the lack of overarching methodological frameworks to guide model development are still major barriers to its wider adoption" (p. 728). When modelling any complex system, a framework or structured method guiding how the model will be developed would help modellers conceptualise a full-system CM and plan how that model would be developed.

S2. *Modellers* code the simulation. i.e., the CM is converted into a computer model.

Robinson (2013) described how the artefacts of the model development process emerge during the life-cycle (Figure 1b), moving from the problem domain, i.e., D1) system description, to the model domain i.e., D2) conceptual model, D3) model design and D4) model code. While advocating engagement throughout the life-cycle Tako and Kotiadis (2015) note model coding “does not provide much room for involving the stakeholders” (p. 557), hence, this may represent a proportion of the project with lower contact between those building the model and key stakeholders. Tako and Kotiadis (2015) recommend identifying a project champion (on the *clients’* side) as a key contact (for this stage and throughout the life-cycle) to liaise with over correctness of model (model validation), however, they do not propose any formal steps to evaluate or review the progress of the model coding throughout this period.

S3. *Modellers* experiment with the simulation model in order to obtain a better understanding of the real world and/or to find solutions to real world problems.

One objective of simulation experimentation is validating the model. Simulation modelling is a highly iterative process (Willemain 1995; Balci 1994; Robinson 2013). As the modeller progresses through the simulation life-cycle loop and begins to experiment with the model, they may be required to repeat some steps (Robinson 2004). During the experimental stage, inability to achieve certain validation targets e.g., replicating real-world outcomes (Law 2009), or recognition of shortcomings in the model’s ability to address the *clients’* problems of interest may prompt revisiting the previous life-cycle steps. *Modellers* must demonstrate to *clients* that the model and its outputs are credible such that its findings can be acted upon (Balci 1994).

The other objective of experimentation is to determine the solution space (Tako and Kotiadis 2015). This process involves understanding how the model can be used to generate insight and address real-world problems (Robinson 2004). How this less tangible experimentation is conducted will depend significantly on the model, the system being modelled and the problem situation in question, hence, there is no standard methodology for doing this. Tako and Kotiadis (2015) advocate debate between modellers and stakeholders as part of this activity to identify the desirable (and feasible) solution space and specific scenarios that satisfy the *clients’* requirements.

S4. *Clients* implement some change or propose some actions as a result of the *modellers* work.

Implementation can take any of three forms; *1)* the model is handed over to the *clients* as a product they can use e.g., for planning weekly schedules, *2)* findings from a simulation study are implemented in reality i.e., a particular solution demonstrated via the model is put into practice or, less explicitly, *3)* learnings are generated which may influence future decision-making (Robinson 2004). These three forms of implementation outline how the *clients* may benefit from the work of the *modellers*. ‘*1)*’ implies the model development is complete but ‘*3)*’, and possibly ‘*2)*’, may emerge during the development of the model earlier in the life-cycle, re-iterating the need to engage stakeholders throughout to maximise the benefits of undertaking the study. Failure to engage stakeholders may result in findings not being acted upon (Brailsford and Vissers 2011; Fone et al. 2003; Young et al. 2009).

2.1 Increasing Model Complexity and Longer Simulation Life-Cycles

Until relatively recently simulation practitioners typically developed models using a single approach or paradigm e.g., static, Monte Carlo, DES, SD or ABM. HS has emerged as computational power has allowed initially used when the complexity of a problem meant the limits of a single simulation method were reached (Brailsford et al. 2018). Now with the hardware and software “barrier greatly reduced hybrid models may be a first choice design, not an afterthought when one approach’s limits are reached” (Jones et al. 2021). As Mustafee, Harper, and Onggo (2020) note in their review of developments in HS, modern simulation packages, specifically AnyLogic (2019), facilitate the development of “mixed models in a single modelling environment” (p. 3141).

There are a variety of definitions for HS in existing literature from different fields, but, broadly “these consider HS as a single simulation method combined with any or multiple of the following; another

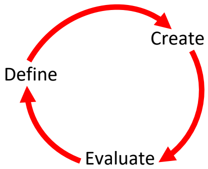

Figure 2: The MVP development process adapted from Münch et al. (2013).

simulation method, an analytic technique or another method that enhances the simulation study" (Jones et al. 2021). The introduction of hybridisation can enable a situation of interest to be assessed from a new perspective or additional, previously neglected features to be captured (Zulkepli and Eldabi 2015). Developing a HS model inevitably adds complexity, requiring broader expertise than the workings of a single simulation method or paradigm. Given the trend to apply simulation to more complex situations, and HS's noted suitability for representing complex systems (Brailsford et al. 2018; Mustafee et al. 2020), complex HS models are increasingly common. For complex HS studies all stages of the life-cycle will likely be long. Model coding particularly may require significant modeller-hours, potentially divided between multiple modellers with specialist expertise in the combined techniques.

2.2 Minimum Viable Product Development

An MVP development approach is not one that develops a minimal product, instead, it is a strategy and process directed toward satisfying the minimum requirements of the client as quickly as possible to ensure value is created (Lenarduzzi and Taibi 2016). This approach allows developers to quickly test their hypothesis as to how the product can deliver value (Ries 2011), minimising the development required to do so. A complete MVP will be a product that satisfies all of the client's requirements. However, the MVP approach promotes the principle of incrementally addressing the client's requirements through a series of development steps, potentially delivering partial products, which can address some but not all requirements, to the client as these emerge for early feedback (Schuh et al. 2018; Ries 2011). The iterative MVP development process of defining requirements, creating a product and evaluating that product's ability to meet the requirements (Münch et al. 2013) is summarised in Figure 2.

The MVP principles are typically associated with start-up companies developing software products and demonstrating their value to customers or investors (Ries 2011). The MVP approach has much overlap with agile software development which, rather than simply tracking overall project maturity, considers incremental developments (e.g., the addition of new features) as key indicators of progress (Poppendieck and Poppendieck 2003), promoting opportunities for feedback from clients and revisions to the project scope as these emerge. These principles are, to the best of our knowledge, not discussed in existing literature in relation to simulation model development. Stakeholder involvement throughout the simulation life-cycle is considered essential for ensuring findings are acted upon (Lowery et al. 1994; Fone et al. 2003; Jun et al. 1999; Eldabi et al. 2007; Brailsford and Vissers 2011; Young et al. 2009). Adopting the MVP principles for the development of complex simulation models would promote engaging stakeholders across the long development process through a series of incremental development stages and generate value from the learning throughout the life-cycle, potentially influencing future thinking and decision making, a benefit of engagement noted by Monks, Robinson, and Kotiadis (2016) and Gogi, Tako, and Robinson (2016).

3 AN MVP APPROACH TO MANAGING THE DEVELOPMENT OF COMPLEX SIMULATION MODELS THROUGHOUT THE SIMULATION LIFE-CYCLE

We propose an MVP simulation life-cycle framework for the development of complex simulation studies, as illustrated in Figure 3. The framework modifies the simulation development process (Robinson 2013) and integrates it into an adapted simulation life-cycle (Robinson 2004) by incorporating an MVP development process i.e., a modified Figure 1b is joined to an adapted Figure 1a by Figure 2. Figure 3 shows three interconnected loops, however, the *green development loop* can be thought of as contained within stage

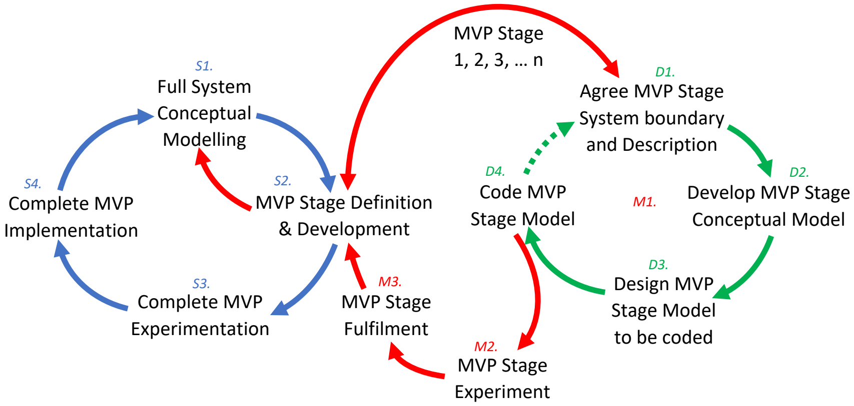

Figure 3: Integrate MVP Lifecycle.

M1 of the *red MVP loop*, which itself can be thought of as contained within stage S2 of the *blue life-cycle loop*. One iteration of the life-cycle loop may involve numerous iterations of the MVP loop (i.e., multiple MVP stages). In turn, any MVP stage may require numerous iterations of the development loop.

We describe progressing through the integrated MVP life-cycle referring back to the life-cycle outlined in Section 2 for reference/comparison. We highlight how our method aids stakeholder engagement in cases where model coding requires significant time. Key to our framework is dividing the study into distinct 'MVP stages' for development (see guidance in Section 4).

3.1 Proceeding Through the MVP Simulation Life-Cycle Framework

S1. *Modellers* works with the *clients* to develop a full system CM.

The objectives here remain the same as those outlined in Section 2 i.e., understand the problem, determine modelling objectives, develop CM inputs, outputs and model content, collect required data, etc (Robinson 2004). When developing a CM of a very complex system *modellers* and *clients* will need to accept early on, that the CM will need revising as the project progresses. As *modellers* and *clients* progress with the simulation study and learn more about the system, the CM will evolve.

S2. *Modellers* works with the *clients* to propose and develop 'MVP stages' i.e., a series of development steps to reach a complete MVP.

This stage involves dividing the development of the full system high-level CM and study requirements into 'MVP stages'. The intention is twofold, firstly, to divide the study and model development process into separate manageable parts and, secondly, to plan and explain, how the model will be built and the study objectives achieved. Section 4 provides guidance on how to define MVP stages.

Consideration should be given to the stages that will follow dividing the model (see Figure 3). How will the MVP stage be developed? How will it be experimented with? And how will stage fulfilment be assessed? Criteria for each of these should be defined here along with any essential design details e.g., inputs required to be received and outputs required to be generated, such that components of the model emerging as MVP stages progress can be joined at a later stage.

Dividing the CM into separate components for development at this point in the life-cycle may prompt revision to the CM (note the addition of the return arrow linking S2 to S1 in Figure 3). Once MVP stages are defined, iterations of the red loop shown in Figure 3, fulfilling the MVP stages, can begin.

M1. For each MVP stage defined, we now develop that stage.

Here, we move from the problem domain to the model domain as the artefacts of the simulation process emerge (Robinson 2013) with iterations of the *green development loop* in Figure 3.

D1. From S2 a system description of the MVP stage is generated. The system boundary and description must be agreed by the *clients* and *modellers* undertaking the development.

D2. From S1, a CM (partial or complete) may exist. Now, the system boundary considered has been reduced to focus specifically on the current MVP stage. The narrower scope reduces the complexity the *modellers* are dealing with. This more manageable complexity may aid further development of the CM for the MVP stage in question, potentially improving the design or adding detail.

D3. A detailed design is developed that proposes how the MVP stage CM will be developed into a computer model. This incorporates details of data structures, components, model execution, etc (Fishwick 1996). The reduced scope of the MVP stage facilitates this activity being manageable in a way it would not be for the full system CM (from S1).

D4. The computer model of the defined MVP stage is coded.

In a collaborative project involving several modellers (or groups of), this stage is important to ensure the future integration of MVP stages. With each iteration of the green loop, findings may emerge which lead to some adjustment of the MVP stage system boundary/description and MVP stage CM (D1 and D2). This may in turn have implications for the other MVP stage definitions. Note, the red arrow connecting S2 to the development loop, M1, is two-way. Some specifics of the design i.e., required inputs and outputs and their format should have been defined in S2 such that MVP components can be integrated. Any change to these features will likely require revisions at S2 and possibly S1. The simulation development process is known to be highly iterative (Willemain 1995; Balci 1994; Robinson 2013) and as a modeller progresses they may be prompted to repeat steps (Robinson 2004). Likely, the greater the complexity of the system being modelled the greater the number of iterations required.

M2. *Modellers* experiment with or demonstrate the MVP stage, possibly integrating into the complete model.

Once an MVP stage has been developed, *modellers* may be able to demonstrate its functionality in some way. This may simply be demonstrating the MVP stage in a test-driven development way (Beck 2003) i.e., demonstrating test cases proposed (at S2) can now ‘pass’ or fulfil specified criteria. In this scenario, the objective is verification that the code is functioning as expected (Law 2009). Other possible demonstrations proposed by *modellers* and *clients* for an MVP stage will depend on the system being modelled. For example, demonstrate a scenario observed in the organisation’s operation to be replicated in simulation, such as a queue building on a specific node in a networks simulation. Alternatively, demonstrate how queue length varies changing input *X*. These demonstrations, and many other examples, may not require the full simulation model to have been developed. Designing MVP stages such that demonstration can be performed after the development stage is completed provides *modellers* an opportunity to engage with *clients* at regular intervals throughout the long simulation development process. Successful demonstration of or experimentation with each MVP stage should give all parties confidence the MVP stages provide an appropriate representation of the part of the system being modelled, implying the complete model under development will similarly. This incorporates validation throughout the development process (Law 2009).

Some experiments may also be possible after the MVP stage has been developed, but before a complete model exists. For example, a complete model may consist of many subsystems. Separate MVP stages may develop those sub-systems. Insight may be generated by experimenting with those subsystems individually before integrating them into a complete model in a later MVP stage. Opportunities for experimentation should be considered by *modellers* and *clients* at S2.

M3. *Clients* and *modellers* jointly determine if the scope of the defined MVP stage has been fulfilled.

The criteria for fulfilment for each MVP stage, and how that will be assessed, should be jointly agreed between *clients* and *modellers* at the definition stage S2. This stage may have significant overlap with the previous i.e., the criteria for fulfilment may be to demonstrate some new model functionality emerging from the MVP stage. However, additional criteria such as the ability to replicate specific quantifiable results or execute computation under a certain time threshold may also be defined.

For a project spanning a long period, this stage provides the opportunity to assess progress against an original project plan. The MVP stage should be assessed to determine if it has fulfilled the criteria defined at S2 and potentially determine any revisions required to the other MVP stage definitions outlined at S2 (and possibly revisions to the full CM proposed at S1) if not. These revisions will be required at S2 before further progression with other MVP stages (i.e., iterations of the red loop - Figure 3). Incremental learning and benefits seen with each MVP stage will keep *clients* engaged with and committed to long projects.

The representation of Figure 3 indicates, via the red loop, projects progress by iterating through MVP stages sequentially. However, if well defined and suitably modular, and modeller resources are available, some MVP stages could proceed in parallel. Some MVP stages may extend the existing model (i.e., need to build on previous stages), others may be developed independently and a further MVP stage may involve integrating some previously developed stages. With the completion of the final MVP stage, the full system CM will have been converted into a computer model.

S3. *Modellers* experiment with the complete simulation model in order to obtain a better understanding of the real world and/or to find solutions to real world problems.

With a complete system model, experimentation can be conducted as the traditional life-cycle described in Section 2.

S4. *Clients* implement some (or further) changes as a result of experimentation with the complete model.

Again this activity is similar to that described in Section 2. As noted at M3, hopefully, several learnings will have been generated during the MVP development process and actions proposed. It may be the case that as the MVP stages progressed, new insight and learnings generated caused the scope of the study to be revised. Hence the resultant model may appear quite different to that originally envisaged. With the project objectives fulfilled, *clients* may wish to propose further work.

4 DEFINING MVP STAGES

Modern software design best practice promotes separating the functionality of a program into modules that can exist independently and interchangeably, containing everything required to perform just one aspect of the complete program's function (Stannett and Dickinson 1990). The term module in software typically refers to *some code*, hence, the development of a module would refer to *the development of some code*. Our framework uses the terminology 'MVP stages' to describe the progression to a complete model. 'Stage' gives greater discretion to the *modellers* and *clients* regarding how they wish to divide their study/model and its development. Our terminology recognises that in the development of a complete model, research into the most appropriate method or experimentation with the code at a certain point may be as or more significant a task (and more time consuming) than the development of the code itself (i.e., referring to Figure 3, D1, D2 and D3 may require significantly greater time than D4 for some MVP stages, and similarly M2).

We adapt the criteria of an effective modular approach proposed by Jones, Keating, and Porter (2001) as guidance to divide our CM development into a set of clear tasks (MVP stages).

1 Define the building blocks of the study, system and model:
- Define clear boundaries, dividing into components that can be plugged in or unplugged (or conducted/developed separately) with little impact on others.
- Facilitate the ability to integrate work undertaken (e.g., analytics and data collection) or model components developed by others. This will speed up model development, enable code reuse, potentially expand the model scope and, in general, shorten the overall time required for the study.
- Define clear objectives and outputs of components to be developed (in separate MVP stages) to enable work undertaken separately or code/model components developed independently to be integrated into the study/model.

2 Promote flexibility in design and use. An MVP development approach should acknowledge the system being modelled, or the requirements of the study may change and you may wish to modify the study and model in future.

- Allow for updates, maintenance and the incorporation of feedback from clients in the design of the study/model.
- Facilitate extending the life and utility of the simulation study and model by incorporating flexibility to edit or add components.

3 Facilitate understanding and improvement of the study design, the model and the system being modelled for both modellers and clients.
- Facilitate comparisons of the model and components of the model/study with alternative designs by clearly defining the scope of components/work and their role in the model/study.
- Facilitate documentation and sharing of code and outcomes by following best practice software engineering and project management methods. Seek to simplify code wherever possible and produce clear documentation of model structure and study findings (e.g., by following the STRESS guidelines (Monks et al. 2019)).
- Consider in the study and model design how modularity could increase the opportunities for collaboration amongst model development groups and opportunities to involve stakeholders.

5 AN EXAMPLE APPLICATION OF THE FRAMEWORK

Jones and Gun (2022) presented a HS methodology to support freight rail operations in the mining industry. The full-system model captured the rail operation, connecting a series of mines to ports, as a hybrid ABM DES model. This was further hybridised with heuristic techniques and experimented with via an ensemble approach. The method identifies destinations for the train fleets to ensure the desired mix of material is transported from mine to port and computes a timetable for the train movements between those destinations. This was a significant project developed over a long time. Here we do not provide extensive details of the development process and collaboration, but instead, use the key high-level milestones described in the Jones and Gun (2022)'s article to retrospectively illustrate our proposed framework.

With a high-level CM proposed (S1 in Figure 3), Jones and Gun (2022)'s model was divided into development stages (S2). The process of developing the complete model can be thought to consist of MVP stages as follows: 1) develop a DES rail network model; 2) introduce train and terminal agents (i.e, hybrid DES and ABM) to create a freight network model including rail network and terminal loading and unloading operations; 3) introduce further hybridisation with analytic methods, here a heuristic for moving the trains between given locations on the network; 4) develop a second heuristic for choosing train destinations; 5) apply an ensemble simulation methodology to identify the optimal configuration of the destinations heuristic and in turn the optimal timetable for the operation.

The MVP stages were developed in the order presented, iteratively progressing through the MVP loop (M1 to M3). Note, each of these was a significant task for which the development time varied from several weeks to several months, collectively resulting in a highly complex model. Each required numerous iterations of the development cycle (D1 to D4).

Further to the development of each MVP stage, there were opportunities to demonstrate progress to the clients (M2), typically supported by visualisation/animation. Numbers corresponding to the above MVP stages, some example MVP demonstrations included: 1) face validation between a representative model and the real network; 2) a basic demonstration of trains moving around the network (with movement instructions manually prescribed); 3) demonstration of trains moving between a given origin and destination via the heuristic and the ability of the system to produce a basic timetable; 4) a demonstration of the trains moving around the network and their next destination being prescribed via the heuristic method. The ability to compute more complex timetables consisting of multiple cycles i.e., each train visiting multiple destinations in turn; 5) the ability to produce high-quality timetables with optimal train destinations.

The demonstration of each complete MVP stage provided the opportunity for feedback from stakeholders and typically specific learnings were generated e.g., efficient traffic movement methods or destination selection strategies learned through observing simulation visualisation or analysing the heuristics logic for assigning a train to a particular destination. With each MVP stage, other learnings that are likely generic to

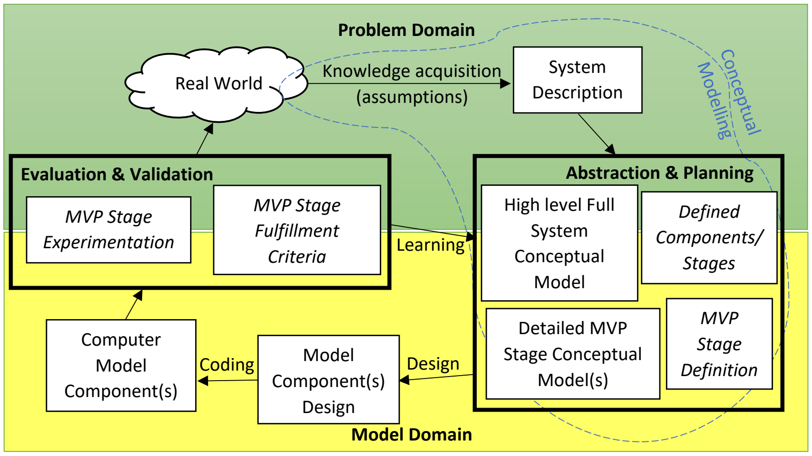

Figure 4: The processes of model abstraction, planning, evaluation and validation our framework promotes and their role in the emergence of the artefacts of the simulation process. Adapted from Robinson (2013).

any project were also generated (e.g., where there are gaps in the clients' data) and perceived wisdom on how the system, or parts of it, are operated were challenged. Modellers and stakeholders collaboratively decided if the MVP stage had been fulfilled (M3) or if further work was required. The completion of all MVP stages resulted in a full system model (S3). From experimentation with that model, it was clear the broad requirements of this system, i.e., to define train destinations and compute timetables for the train movements between those destinations, were satisfied, thus representing a complete life-cycle loop (S4).

6 DISCUSSION

The cross-disciplinary hybrid modelling framework we have presented (Type E - see (Mustafee, Harper, and Onggo 2020)) aims to aid clearer thinking about a model's structure and its development (Kreuger et al. 2016) regardless of the software package or programming language used. It is particularly suited to complex HS development, as HSs are often modular in nature (Kreuger et al. 2016). The boundaries of hybridisation are intuitive divides within the full system CM. For many HS models, the modular structure will naturally generate a set of MVP stages within our framework, though these may be decomposed further. Note that in our example (Section 5) the MVP stages align with three of the five possible component types of Jones et al. (2021)'s modelling frame representation, which indicates how different HS components are combined to address a problem. The first two MVP stages fall within 1) HS, the following two fall within 2) HS and analytic model and the final falls within 4) the hybrid experimental approach. Defining the modelling frame can aid the definition of MVP stages. Planning the stages of model development, as our framework, encourages modellers to clearly define the purpose of introducing hybridisation. Specific test criteria can be proposed, within our framework, to evaluate if those benefits are realised when the hybridisation is introduced. The structured method guiding model development we propose can help modellers conceptualise a full-system CM and consider how hybridisation can be used to maximum effect.

Figure 4 illustrates how the processes integral to our framework are expected to impact the emergence of the model artefacts. With each MVP stage modellers' expertise in the system increases, hence, their ability to make recommendations on how to model the system increases. The framework provides a possible method for abstracting the real world and planning the model development process, bridging the gap between the problem domain and model domain (Robinson 2013). The development of each stage provides opportunities to engage closely with stakeholders and ensures transparency of assumption, simplifications and limitations throughout the modelling process, a key step in the process of validation (Law 2009).

Our framework aims to promote testing and learning from the behaviour of the model (Kreuger, Qian, Osgood, and Choi 2016). This aligns with an agile development philosophy (Poppendieck and Poppendieck

2003) and provides modellers with the opportunity to engage clients in model verification and validation as the model is being developed (Law 2009). Modellers can use testing and demonstrations as opportunities to evaluate development progress and facilitate learning as the simulation project develops (Lowery et al. 1994; Jun et al. 1999; Eldabi et al. 2007; Brailsford and Vissers 2011; Fone et al. 2003; Young et al. 2009). Our framework recognises that understanding of the highly complex system being modelled will develop as model development progresses. A key part of the CM process is understanding the system to be modelled and vital to this is collaboration with stakeholders (Kotiadis, Tako, and Vasilakis 2014). It is expected that our framework will promote incorporating learning from stakeholder feedback as the model is developed. This may mean the CM is revised as the model develops, but, it ensures the final model produced is the most appropriate for addressing the problem at hand. Figure 4 indicates how learning generated with the progression of each MVP stage feeds back to the CM design.

Through a case study, we have attempted to demonstrate how, via our framework, a larger project can be decomposed into smaller components. It is hoped that developing a simulation model via our proposed framework facilitates incremental development (Kreuger, Qian, Osgood, and Choi 2016) enabling agile software development techniques to be adopted (Poppendieck and Poppendieck 2003), the typical method used for developing an MVP. An agile method has many advantages well documented in software development literature. It promotes short iterative development cycles (Boehm and Turner 2005) and a highly collaborative and evolutionary approach (Moniruzzaman and Hossain 2013). Agile development actively involves users to establish, prioritise, and verify requirements (Petersen and Wohlin 2009). It emphasises rapid delivery of a product to the clients, prioritising key features and involving clients to ensure regular feedback (Petersen and Wohlin 2009). This typically results in a reduced development time compared to traditional development approaches, with less waste (Moniruzzaman and Hossain 2013). Further, the modular nature of agile facilitates teams to 'self organise' as the tasks to be achieved are clearly defined and easily divided (Leffingwell 2010). Our framework defines MVP stages. When these are defined, an order in which these need to be developed, and those which can be developed in parallel, will be clear. Similarly, the resources and modeller expertise required for each stage will be clear and estimating the time required for completion of stages will be much easier than estimating the time for the development of a full system model. Hence, our framework could aid project planning and management.

The framework provides a tool modellers may use to communicate to stakeholders how a model will emerge over time and could aid them in proposing a timeline. Modellers could use our framework to communicate how stakeholder requirements will be met as the model develops. Used in parallel with existing methods for communicating and documenting models (e.g., Jones et al. (2021)'s hybrid frame representation, methods for illustrating single paradigm models (Coyle 1997; Oscarsson and Moris 2002; Triebig and Klügl 2009) and the STRESS (Strengthening The Reporting of Empirical Simulation Studies) guidelines (Monks et al. 2019)) could provide a comprehensive tool kit to communicate and document a complex model design, how it will address the questions of interest and how it will be developed.

We developed the proposed MVP-framework retrospectively. Although we believe our experience (Section 5) indicates the potential value, we acknowledge this does not constitute validation of its efficacy. We hope that other simulation practitioners will see the value of our framework, utilise it in their own practice and report their experience. Future work could attempt to compare our framework with others in existing literature by defining and measuring metrics relating to key aspects of the model development process such as stakeholder satisfaction and quality of the model produced.

7 CONCLUSION

We present a novel framework which aims to aid the development of highly complex simulation studies by aiding clear thinking about the study structure and deconstructing the study into components and defined stages of development. The framework is particularly suited to HS studies. MVP development principles are incorporated into the simulation life-cycle. Our framework provides a structured method designed to help modellers conceptualise a full-system CM and plan how that model will be developed into a computer model. In turn, our framework could be used to aid modellers in conveying that plan to stakeholders.

ACKNOWLEDGMENTS

This work was supported by the Rio Tinto Centre for Mine Automation & the Australian Centre for Field Robotics, University of Sydney.

REFERENCES

AnyLogic 2019. "AnyLogic: Simulation Modeling Software Tools & Solutions for Business". https://www.anylogic.com/ accessed 12th June 2019.

Balci, O. 1994. "Validation, Verification, and Testing Techniques Throughout the Life Cycle of a Simulation Study". *Annals of Operations Research* 53(1):121–173.

Beck, K. 2003. *Test-Driven Development: By Example*. Addison-Wesley Professional.

Boehm, B., and R. Turner. 2005. "Management Challenges to Implementing Agile Processes in Traditional Development Organizations". *IEEE Software* 22(5):30–39.

Brailsford, S., T. Eldabi, M. Kunc, N. Mustafee, and A. F. Osorio. 2018. "Hybrid Simulation Modelling in Operational Research: A State-Of-The-Art Review". *European Journal of Operational Research* 278(3):721–737.

Brailsford, S., and J. Vissers. 2011. "OR in Healthcare: A European Perspective". *European Journal of Operational Research* 212(2):223–234.

Chwif, L., M. R. P. Barretto, and R. J. Paul. 2000. "On Simulation Model Complexity". In *Proceedings of the 2000 Winter Simulation Conference, edited by J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick.*, Volume 1, 449–455. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Coyle, R. G. 1997. "System Dynamics Modelling: A Practical Approach". *Journal of the Operational Research Society* 48(5):544–544.

Eldabi, T., M. Balaban, S. Brailsford, N. Mustafee, R. E. Nance, B. S. Onggo, and R. G. Sargent. 2016. "Hybrid Simulation: Historical Lessons, Present Challenges and Futures". In *Proceedings of the 2016 Winter Simulation Conference, edited by T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick*, 1388–1403. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Eldabi, T., R. J. Paul, and T. Young. 2007. "Simulation Modelling in Healthcare: Reviewing Legacies and Investigating Futures". *Journal of the Operational Research Society* 58(2):262–270.

Eldabi, T., A. A. Tako, D. Bell, and A. Tolk. 2019. "Tutorial on Means of Hybrid Simulation". In *Proceedings of the 2019 Winter Simulation Conference edited by N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son.*, 33–44. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Fishwick, P. A. 1996. *Simulation Model Design and Execution: Building Digital Worlds*. Prentice Hall PTR.

Fone, D., S. Hollinghurst, M. Temple, A. Round, N. Lester, A. Weightman, K. Roberts, E. Coyle, G. Bevan, and S. Palmer. 2003. "Systematic Review of the Use and Value of Computer Simulation Modelling in Population Health and Health Care Delivery". *Journal of Public Health* 25(4):325–335.

Gogi, A., A. A. Tako, and S. Robinson. 2016. "An Experimental Investigation into the Role of Simulation Models in Generating Insights". *European Journal of Operational Research* 249(3):931–944.

Jones, J., B. Keating, and C. Porter. 2001, nov. "Approaches to Modular Model Development". *Agricultural Systems* 70(2):421–443.

Jones, W., and P. Gun. 2022. "Train Timetabling and Destination Selection in Mining Freight Rail Networks: A Hybrid Simulation Methodology Incorporating Heuristics". *Journal of Simulation* 3:1–14.

Jones, W., K. Kotiadis, J. O'Hanley, and S. Robinson. 2021. "Aiding the Development of the Conceptual Model for Hybrid Simulation: Representing the Modelling Frame". *Journal of the Operational Research Society*:1–19.

Jun, J. B., S. H. Jacobson, and J. R. Swisher. 1999. "Application of Discrete-Event Simulation in Health Care Clinics: A Survey". *Journal of the Operational Research Society* 50(2):109–123.

Kotiadis, K., A. A. Tako, and C. Vasilakis. 2014. "A Participative and Facilitative Conceptual Modelling Framework for Discrete Event Simulation Studies in Healthcare". *Journal of the Operational Research Society* 65(2):197–213.

Kreuger, L. K., W. Qian, N. Osgood, and K. Choi. 2016. "Agile Design Meets Hybrid Models: Using Modularity to Enhance Hybrid Model Design and Use". In *Proceedings of the 2016 Winter Simulation Conference, edited by T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick*, 1428–1438. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Kreutzer, W. 1986. *System Simulation Programming Styles and Languages*. Boston, MA: Addison-Wesley Longman Publishing Co., Inc.

Law, A. M. 2009. "How to Build Valid and Credible Simulation Models". In *Proceedings of the 2009 Winter Simulation Conference, edited by M.D. Rossetti and R.R. Hill and B. Johansson and A. Dunkin and R.G. Ingalls*, 24–33. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Leffingwell, D. 2010. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley Professional.

Lenarduzzi, V., and D. Taibi. 2016. "MVP Explained: A Systematic Mapping Study on the Definitions of Minimal Viable Product". In *Proceedings of the 2016 42nd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 112–119. Institute of Electrical and Electronics Engineers, Inc.

Lowery, J., B. Hakes, W. Lilegdon, L. Keller, K. Mabrouk, and F. McGuire. 1994. "Barriers to Implementing Simulation in Health Care". In *Proceedings of the 1994 Winter Simulation Conference, edited by J.D. Tew and S. Manivannan and D.A. Sadowski and A.F. Seila*, 868–875. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Moniruzzaman, A., and D. S. A. Hossain. 2013. "Comparative Study on Agile Software Development Methodologies". *arXiv preprint arXiv:1307.3356*. https://arxiv.org/abs/1307.3356 accessed 30th August 2022.

Monks, T., C. S. M. Currie, B. S. Onggo, S. Robinson, M. Kunc, and S. J. E. Taylor. 2019. “Strengthening the Reporting of Empirical Simulation Studies: Introducing the STRESS Guidelines”. *Journal of Simulation* 13(1):55–67.

Monks, T., S. Robinson, and K. Kotiadis. 2016. “Can Involving Clients in Simulation Studies Help Them Solve Their Future Problems? A Transfer of Learning Experiment”. *European Journal of Operational Research* 249(3):919–930.

Münch, J., F. Fagerholm, P. Johnson, J. Pirttilahti, J. Torkkel, and J. Jäarvinen. 2013. “Creating Minimum Viable Products in Industry-Academia Collaborations”. In *International Conference on Lean Enterprise Software and Systems*, 137–151. Berlin, Heidelberg: Springer.

Mustafee, N., S. Brailsford, A. Djanatliev, T. Eldabi, M. Kunc, and A. Tolk. 2017. “Purpose and Benefits of Hybrid Simulation: Contributing to the Convergence of its Definition”. In *Proceedings of the 2017 Winter Simulation Conference, edited by W.K.V. Chan and A. D’Ambrogio and G. Zacharewicz and N. Mustafee and G. Wainer and E. Page*, 1–15. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Mustafee, N., A. Harper, and B. S. Onggo. 2020. “Hybrid Modelling and Simulation (M&S): Driving Innovation in the Theory and Practice of M&S”. In *Proceedings of the 2020 Winter Simulation Conference, edited by K.H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing*, 3140–3151. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Oscarsson, J., and M. U. Moris. 2002. “Documentation of Discrete Event Simulation Models for Manufacturing System Life Cycle Simulation”. In *Proceedings of the 2002 Winter Simulation Conference, edited by E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes*, Volume 2, 1073–1078. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Petersen, K., and C. Wohlin. 2009. “A Comparison of Issues and Advantages in Agile and Incremental Development Between State of the Art and an Industrial Case”. *Journal of Systems and Software* 82(9):1479–1490.

Pidd, M. 1988. *Computer Simulation in Management Science*. 2nd ed. New York, NY: John Wiley & Sons, Inc.

Poppendieck, M., and T. Poppendieck. 2003. *Lean Software Development: An Agile Toolkit*. Boston, Massachusetts: Addison-Wesley.

Ries, E. 2011. *The Lean Startup: How Today’s Entrepreneurs use Continuous Innovation to Create Radically Successful Businesses*. New York, New York: Crown.

Robinson, S. 2004. *Simulation: The Practice of Model Development and Use*. Chichester: John Wiley & Sons.

Robinson, S. 2013. “Conceptual Modeling for Simulation”. In *Proceedings of the 2013 Winter Simulation Conference, edited by R. Pasupathy and S.-H. Kim and A. Tolk and R. Hill and M.E. Kuhl*, 377–388. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Robinson, S. 2020. “Conceptual Modelling for Simulation: Progress and Grand Challenges”. *Journal of Simulation* 14(1):1–20.

Robinson, S., C. Worthington, N. Burgess, and Z. J. Radnor. 2014. “Facilitated Modelling with Discrete-Event Simulation: Reality or Myth?”. *European Journal of Operational Research* 234(1):231–240.

Schuh, G., C. Dulle, and S. Schloesser. 2018. “Agile Prototyping for Technical Systems–Towards an Adaption of the Minimum Viable Product Principle”. *DS 91: Proceedings of NordDesign 2018, Linköping, Sweden*:1–24.

Stannett, M., and S. Dickinson. 1990. *Modular Software Design*. Chislehurst, London: Chartwell-Bratt.

Tako, A. A., T. Eldabi, P. Fishwick, C. C. Krejci, and M. Kunc. 2019. “Panel - Towards Conceptual Modeling for Hybrid Simulation: Setting the Scene”. In *Proceedings of the 2019 Winter Simulation Conference edited by N. Mustafee, K.H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son.*, 1267–1279. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Tako, A. A., and K. Kotiadis. 2015. “PartiSim: A Multi-Methodology Framework to Support Facilitated Simulation Modelling in Healthcare”. *European Journal of Operational Research* 244(2):555–564.

Triebig, C., and F. Klügl. 2009. “Elements of a Documentation Framework for Agent-Based Simulation Models”. *Cybernetics and Systems* 40(5):441–474.

Willemain, T. R. 1995. “Model Formulation: What Experts Think About and When”. *Operations Research* 43(6):916–932.

Young, T., J. Eatock, M. Jahangirian, A. Naseer, and R. Lilford. 2009. “Three Critical Challenges for Modeling and Simulation in Healthcare”. In *Proceedings of the 2009 Winter Simulation Conference, edited by M.D. Rossetti and R.R. Hill and B. Johansson and A. Dunkin and R.G. Ingalls*, 1823–1830: Institute of Electrical and Electronics Engineers, Inc.

Zulkepli, J., and T. Eldabi. 2015, dec. “Towards a Framework for Conceptual Model Hybridization in Healthcare”. In *Proceedings of the 2015 Winter Simulation Conference, edited by L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossett*, 1597–1608. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

AUTHOR BIOGRAPHIES

WILLIAM JONES is a Senior Simulation Consultant at GHD Advisory. This research was undertaken while in his previous role as a Research Associate at the Rio Tinto Centre for Mine Automation, Australian Centre for Field Robotics, University of Sydney, AU. He earned an EngD in Systems Engineering (2019) from The University of Bristol, UK. His research interests include simulation and optimisation of multi-agent systems, automation and the process of model development. His e-mail address is william.jones@ghd.com.

PHILIP GUN is a Research Associate at the Rio Tinto Centre for Mine Automation, Australian Centre for Field Robotics, University of Sydney, AU. He received a PhD in 2021 from the University of Sydney. His research focus in on the coordination, motion planning, optimisation, and simulation of multi-agent systems, in particular autonomous vehicles. His e-mail address is philip.gun@sydney.edu.au. His website is https://www.sydney.edu.au/engineering/about/our-people/academic-staff/philip-gun.html.

MEHDI FOUMANI is a Senior Research Fellow at the Rio Tinto Centre for Mine Automation, Australian Centre for Field Robotics, University of Sydney, AU. He holds a Ph.D. degree in Operations Research from Monash University, AU. His research interest include optimisation and operations research, applied to manufacturing, energy efficiency, mining, transport and logistics industries. His e-mail address is mehdi.foumani@sydney.edu.au. His website is https://www.sydney.edu.au/engineering/about/our-people/academic-staff/mehdi-foumani.html.