Proceedings of the 2024 Winter Simulation Conference
H. Lam, E. Azar, D. Batur, S. Gao, W. Xie, S. R. Hunter, and M. D. Rossetti, eds.

ON GUIDING SIMULATION MODEL REUSE FROM THE CONCEPTUAL MODELING STAGE

Xiaoting Song[1], Maurizio Tomasella[1], Lazuardi Almuzaki[1], Jamal Ouenniche[1], and Silvia Padrón[2]

[1]Management Science and Business Economics, University of Edinburgh Business School, 29 Buccleuch Place, EH8 9JS, Edinburgh, Scotland, UK
[2] Information, Operations, and Management Science, TBS Business School, 1 Place Alphonse Jourdain, 31068, Toulouse, FRANCE

ABSTRACT

This study proposes a five-stage decision-making process to streamline the reuse of simulation models from as early as the conceptual modeling stage of a new study. The stages assist modelers in selecting pre-existing conceptual and/or formal models, testing their suitability for reuse, selecting model components deemed reusable, adapting them to the new modeling requirements, integrating them into a 'final' conceptual model and, prior to its computer coding, carrying out various steps of conceptual model validation along the way. Key novelties are: a structured approach to model reuse, an emphasis on validation steps, and the integration of practical tools. We discuss the advantages from following such a process based on evidence from a recent project that looked at enhancing aircraft turnaround processes.

1 INTRODUCTION

With the growing sophistication of simulation models, the advantages of leveraging existing models —significant reduction in development time and cost— have become increasingly evident (Robinson 2004). Despite the potential benefits, the practice of simulation model reuse has yet to achieve widespread adoption (Balci et al. 2008; Balci et al. 2011; Balci et al. 2017).

Major challenges include the need to find a balance between model reuse and the necessity for bespoke model development (Robinson 2004), and to ensure the validity of models once repurposed and applied beyond their original context. Modeling *objectives*, model *inputs* and *outputs*, model *content* (scope and level of detail), and modeling *assumptions* and *simplifications*—a 'list-of-six' which many consider equivalent to the *conceptual model* (Robinson 2015)—of the current study ought to undergo rigorous evaluation vs any models considered for potential reuse, so to mitigate the risk of producing invalid models and results (Pidd and Carvalho 2006; Peng et al. 2016). However, the inherently context-specific nature of most simulation models further complicates matters. Adapting such models for use in new domains (or new uses in the same domain) requires considerable expertise, and existing methodologies meant to support such adaptations have been criticized as underdeveloped (Yilmaz and Ören 2004). In addition, the intricate interdependencies within many models significantly challenge their decomposition into components suitable for reuse (Overstreet et al. 2002).

Still, given the resource-intensive nature of simulation studies, pursuing effective and efficient model reuse may bear the much sought-after significant cost and time savings and, most importantly, enhance the models' utility in decision support and scientific exploration (Balci and Ormsby 2007). Therefore, in this paper, we attempt to demystify the process of simulation model reuse. We aim to assist modelers in determining whether complete or partial reuse of existing models is appropriate for a given problem. Additionally, we seek to demonstrate the effective implementation of such reuse in the form of a structured decision-making process.

979-8-3315-3420-2/24/$31.00 ©2024

2 RELATED WORKS

The simulation community has long been interested in reusability (Sargent et al. 1986), with the topic frequently reemerging (Kovács et al. 1999). Reese and Wyatt (1987) discuss how reusing model components can improve simulation software development by increasing productivity, maintainability, and model quality. They suggest model reuse to be promising in domains such as manufacturing, where adoption of *architectural* and *design patterns* — see Coad (1992) for a well-known introduction—are prevalent.

From the 1980s to the present, model reuse has completed four levels of evolution: subprogram-level, component-level, platform-level, and across-system-level reuse (Liu et al. 2016). Many developments in the area cut across these levels. For instance, a preoccupation with federated modeling and the High-Level Architecture (HLA) in military applications (US Department of Defence 1996) gave us a standardized framework for communication and data exchange between different simulation systems, but also facilitated the modularity of simulation model components, and therefore model reuse (Dahmann 1997; Nance 1999). Robinson et al. (2004) noticed that model reuse has often been seen as a near-synonym for software reuse. Perhaps, the very idea that simulation modelers might want to reuse their own or someone else's models (components) may have led very naturally to focusing mostly on the reuse of the software implementing such models (components) — the *formal* model discussed in Oral and Kettani (1993)—, leaving reuse of *conceptual* models in the background. This conjecture couldn't be possibly further from the truth. An early example is that of Wyatt (1990), which introduced a framework for producing reusable model components through graph-based pictorial representations. The framework promotes composability, by instantiating distinct sub-model types for different purposes, which are then combined into larger models.

Object-oriented modeling (OOM) and Object-oriented Analysis and Design (OOAD), which originated in the discourse on software lifecycle and systems analysis and design, have both contributed over the years to more widespread adoption of architectural and design patterns across many disciplines and fields of application and, with it, further highlighted the central role of conceptual modeling. Fowler (2018), arguably the most well-known book on the Unified Modeling Language (UML) — the *de facto* graphical standard for OOM—, talks about using UML in different ways: as a *sketch*, a detailed design blueprint (for a programmer to code up), or directly as a programming language. In the latter use, UML modeling elements map directly to software components, whilst when used as a sketch, they focus on the concepts of the domain to be analyzed: "… Here, we aren't talking about software elements so much as we are building a vocabulary to talk about a particular domain." (Fowler 2018).

Pace (2000) pointed out that the reuse of simulation models should be more concerned with the reuse of conceptual models, as these form the foundation upon which the entire simulation endeavor is built. Ensuring the 'list-of-six' is well specified at the outset and maintained throughout any simulation study has been argued as one of the major enablers of clear communication, alignment with stakeholders, model validity, as well as accuracy, efficiency, and cost-effectiveness of simulations (Robinson et al. 2010). Our work is therefore not the first to address the issue of reusability and modularity at the conceptual modeling level. Additional early examples include the Integration DEFinition for information modeling (IDEF1X) methodology (US Department of Defence 1993) and the Conical Methodology (CM) (Nance 1994). The latter helps to improve the understandability and maintainability of 'knowledge' models through hierarchical development and standardized documentation (not surprisingly, the CM is object-oriented). Still, recent authoritative commentary in Robinson (2020) demonstrated that "Conceptual modeling and model reuse, interoperability, and composability" and "Conceptual modeling and model validation" are somewhat less explored themes (the author categorizes them amongst the *new themes*, especially if compared to "Identifying, adapting, and developing conceptual modeling frameworks" and "Adopting/developing appropriate model representation methods". Hence the need for and focus of our work are presented here.

Ray Paul's viewpoint in Robinson et al. (2004) — a paper documenting a panel discussion on model reuse held at the UK OR Society's Simulation Workshop in March 2002—argues that "model reuse is essentially dependent on trust. If a modeler cannot trust a model then surely they cannot reuse it. It seems to follow that for a modeler to reuse a model, then the modeler must build trust, a process that might take more time than building the model from the start" (Robinson et al. 2004). Paul argues that especially in the era of the World Wide Web, simulation model development tools "that allow for fast model building and quick and easy experimentation" should be adopted, such as the minimalist G^2R^3 framework advocated by himself, which will help *web-enabled* simulation analysts "to assemble rather than build" models. However, the framework advocates for the assembly of software code, instead of conceptual models.

El Haouzi et al. (2008) defined simulation model reusability as the leveraging of generalizable knowledge from models representing systems with shared characteristics, rather than examining each system individually and building unique components. Methodologies such as the Conical Methodology do not provide simulation practitioners with detailed guidance on how they may extract generalized knowledge from existing models. This seems essential for model reuse attempts to be grounded on what is gathered through such a process of learning about, understanding, and building trust in the models available at hand.

From a simulation practitioner's viewpoint, the primary challenges seem to lie in: (i) the systematic collection and appropriate documentation, over time, of potentially reusable simulation resources (both software and conceptual); (ii) the focused selection, from said collection, of the subset of 'most promising' conceptual models for prompt reuse in a new simulation study; (iii) the extraction of reusable elements from the selected models, and their customization to the new needs; (iv) the assembly of the reused, customized, partial conceptual models into the final conceptual model for the study at hand; (v) the assurance of the conceptual validity (hence credibility, trust) of the models resulting from (iii) and (iv). These are all decisions and activities supported by many of the methodologies, methods, and techniques that have been developed recently by the simulation community—e.g. Tako and Kotiadis (2015), Jones et al. (2022).

Perhaps the new knowledge that this body of works has made available should be mapped onto the fundamental steps of a generic template of model reuse, in a way to resolve the issue of the lack of detail found in existing model reuse methodologies which, this way, would become more relevant to practitioners and scholars alike. This is exactly what we pursue in the following. We develop a high-level design of a decision-making process through which said fundamental steps can be followed, executed, and documented, and show where and how in the process some of the most relevant simulation methodologies, methods, and techniques can be integrated.

3 HIGH-LEVEL DESIGN OF A DECISION-MAKING PROCESS FOR MODEL REUSE

We hereby propose a decision-making process to help simulation modelers navigate the various phases in which any attempt at model reuse can be systematically organized. The focus is on conceptual modeling and validation. Figure 1 shows the process in detail, and its vertical organization in the following five stages (top to bottom): ***Identify***, ***Set***, ***Assess***, ***Conceptualize***, and ***Code***, which we now discuss.

Our flowchart notation is inspired by BPMN (Business Process Modeling and Notation). The terminating nodes in Stages 1 and 5 (gray background) represent the inputs and outputs of the process respectively. The core of our process (Stages 1–4, on which this description focuses) maps directly onto the *modeling* and *conceptual modeling validation* activities of the stages in a simulation study as presented in Figure 1 of Brailsford et al. (2019). Stage 5 involves the coding of the final conceptual model, now validated, into a formal model, and its related verification. $S(\cdot)$ refer to the problems/systems of interest, whilst $S_{\cdot}^{M}(\cdot)$ point at the related models. Conceptual and formal (software) models are denoted as $S_{c}^{M}(\cdot)$ and $S_{f}^{M}(\cdot)$ respectively. Textual and graphical representations of conceptual models are denoted as $S_{c,t}^{M}(\cdot)$ and $S_{c,g}^{M}(\cdot)$ respectively ($S_{c}^{M}(\cdot) = S_{c,t}^{M}(\cdot) \cup S_{c,g}^{M}(\cdot)$). Everywhere along the flow, $AS-WAS$ ($AS-IS$) in brackets refer to problems/systems or models from a past (the current) project/study. The diamond-shaped nodes in Stage 3 represent the two main decision points. All nodes in Stages 2 and 4 are instead either inputs ($S_{c}^{M}(AS-WAS)$ and $S_{f}^{M}(AS-WAS)$) or intermediate outputs (all the others). Circles represent connectors of the 'AND' type ('parallel gateways', in BPMN terminology). Solid (dashed) lines with arrows represent the workflow in the process that relates to modeling (validation) activities. Dashed horizontal lines without arrows simply divide elements belonging to two contiguous stages.

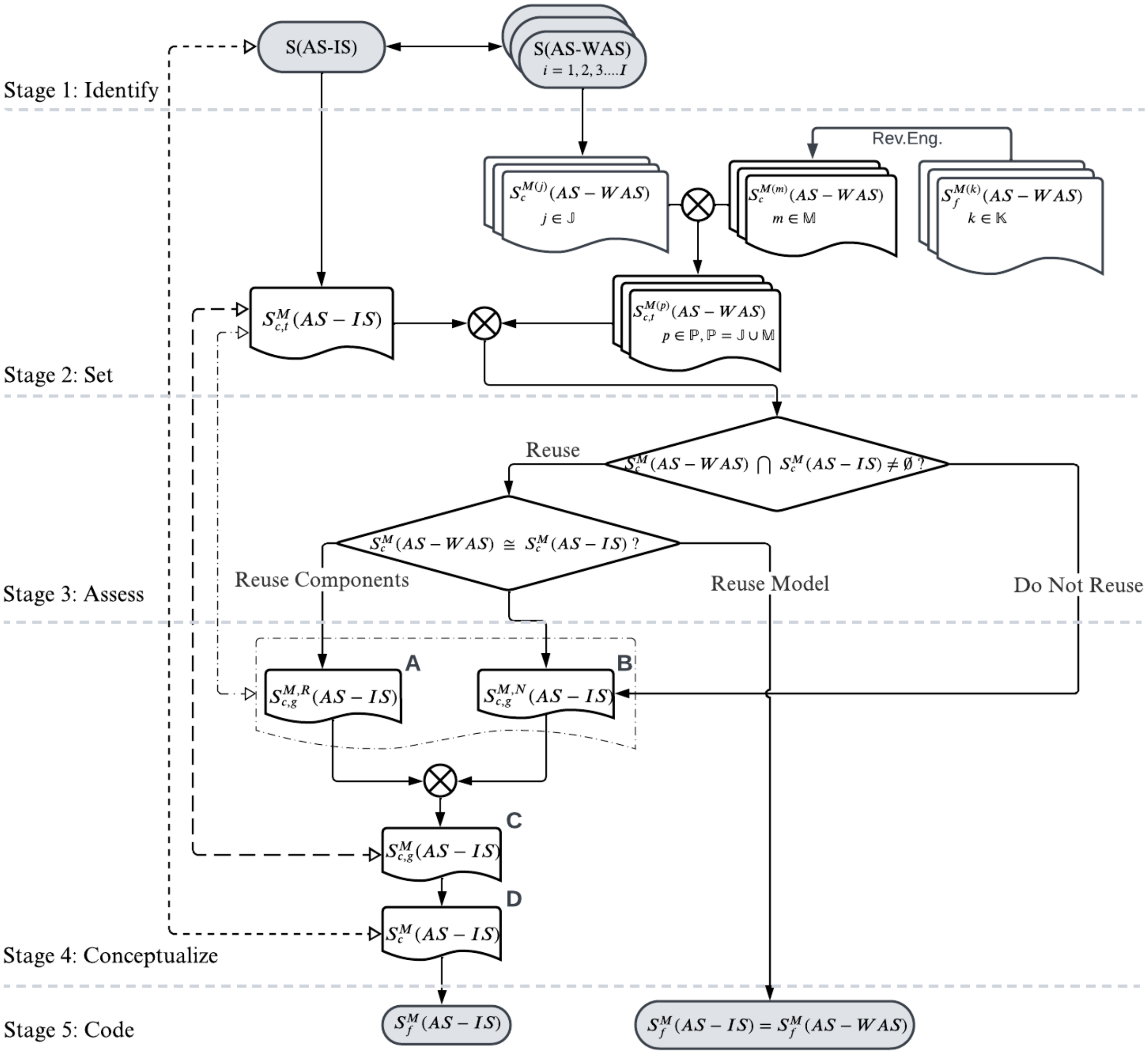


Figure 1: A high-level design for simulation model reuse.

In Stage 1, the modeler starts with acknowledging the main elements of the current project/study—the ‘finding out’ activity in Soft Systems Methodology (SSM) (Checkland and Scholes 1999) or the ViPlan Methodology (Harwood 2021)—, the problem/system of interest, and any other relevant contextual information, including the viewpoints (or *worldviews*) from which it is being looked at. With this phase of *problem structuring* completed to a large extent, relevant previous projects (indexed as $i = 1, \dots, I$) and related problems/systems can be identified, of which either formal or conceptual models and accompanying documentation are available to (or can be promptly gathered by) the modeler. These are likely to cover different contexts and worldviews than those in the current study, have probably been carried out by a wide range of different individuals, teams, and perhaps organizations, documented through different modeling languages and conventions, and implemented in different tools (including software). Occasionally, experience from these former engagements may have been documented in academic articles.

Real-world problem situations are inherently complex, multifaceted, and embedded in specific organizational (as well as social, economic, environmental, technological, political, etc.) contexts. Direct comparisons are unlikely to be meaningful unless an appropriate methodological approach is taken. The mature area of Problem Structuring Methods (PSMs) (Mingers and Rosenhead 2004) may assist the modeler in picking from a wide catalog of approaches. Focusing on a small set of PSMs, over a prolonged time, will help the modeler to document their projects in a way that the comparison in Stage 1 may be carried out more efficiently as time passes, thus resulting in a more credible selection of past projects to be taken forward to Stages 2–4. Many instances exist where such guidance has been incorporated into frameworks to aid decision-making in simulation endeavors. For instance, Tako and Kotiadis (2015) blends ideas from SSM and discrete event simulation (DES) to propose a framework that facilitates DES by boosting stakeholder participation.

In Stage 2 we set everything ready for decision-making in Stage 3, where the modeler will choose whether to reuse existing models and to what extent (i.e. whole model vs components), or instead build a new model altogether, to solve/analyze the current problem/system. Stage 2 produces two main intermediate outputs. The first (left-hand side of Figure 1) is a textual version of the conceptual model for the current problem. The list-of-six cited in our introduction above constitutes a fairly standard approach that is followed by many. The second is a selected subset of conceptual models from previous work and/or literature that are deemed relevant for reuse. Their representation ought to possess some level of homogeneity, to streamline decision-making in Stage 3. Ideally, a textual version of the conceptual model for them is available, even better if following the same notation, conventions, and methods of the conceptual model for the current problem/system. Graphical versions of the conceptual models 'from the past' will often be available. For every past work/project (index i) considered, several conceptual models (j) may be available—e.g. representing different system configurations studied—, as well as several formal models (k). Some of the formal models may have no conceptual model counterpart: in such cases, it can prove useful for the modeler to reverse engineer the related conceptual models (m). All available textual versions of said conceptual models (p) are then collated for use, in conjunction with the model for the current problem/system, in Stage 3. The explicit recommendation that textual versions of conceptual models may be particularly helpful at this stage should not come as a surprise: "Similar to drawing, but in the cultural processes of written language, most phenomena to be modeled are best described in writing first. Only then, can the writing be analyzed and cross-checked against the sorts of components in a model." (Fishwick and Mustafee 2019). Writing is indeed modeling —as in SSM—or an important early stage of it —as in System Dynamics (SD) (Forrester 2013).

In Stage 3, detailed reasoning about the content of all textual conceptual models available at this point will highlight what components are indeed suitable for reuse, and what shall be modeled from the outset (or, though perhaps unlikely, that one of the existing models may be reused as a whole). The modeler may attempt to semi-automate this reasoning step, at least to some extent, grounding on the capabilities of more recent developments such as in Large Language Models or more established ones such as Case-Based Reasoning (Guo et al. 2024).

In Stage 4, the conceptualization of the current problem/system may be then completed. Focusing on the more likely case of partial model reuse, the inputs to Stage 4 (and the outputs of Stage 3) are a textual description of both the components that are indeed reused (A in Figure 1) and those that were part of the original conceptual model of the current problem/system but have not found any suitable component from the available knowledge base (B). Some rework of A may be needed, as well as component-level validation (dashed line cutting across onto Stage 2). Simulation modelers may then want to transform the (now validated) component-level textual models into some related graphical representations of both A and B. These will then be integrated into an overall graphical conceptual model of the current problem/system (C), on which to carry out further validation (additional dashed line across to Stage 2). The ensemble of textual and graphical conceptual model components of the current problem/system (D) may be then validated vs. the perception of the same that was built initially (dashed line linking on to Stage 1). With both conceptual modeling and related validation completed successfully, model coding (Stage 5) may begin.

4 EMPIRICAL EVIDENCE

4.1 Research Context and Problems/Systems of Interest

We have developed and tested the decision-making process in Figure 1 over the past six years, in over a dozen projects (research and consultancy). All such projects have looked at the same area of inquiry— airport airside operations management. Our work has explored a number of viewpoints, including those of the airport operator, the air traffic services provider, the airline, and the ground handling service providers. Some of these studies have been published (Cattaneo 2018; Tomasella et al. 2019; Gök et al. 2020; Saggar et al. 2021; Saha et al. 2021; Gök et al. 2023). All of our simulation models have been DES/ABS (Discrete-Event Simulation/Agent-Based Simulation) hybrids. The first studies (Cattaneo 2018; Tomasella et al. 2019) added an agent-based dimension (coded in Anylogic) to a model from an unpublished work of ours from several years prior (coded in Arena). In response to the requirements of each specific project, we hybridized some of our simulation models with Metaheuristics (Tomasella et al. 2019; Gök et al. 2023), Reinforcement Learning (Saha et al. 2021) and Constraint Programming (Gök et al. 2023).

Reducing operational costs by minimizing flight delays has always been a concern for airlines. The aircraft turnaround process significantly contributes to these delays. In brief, this encompasses the servicing of an aircraft between its arrival (Scheduled In-Blocks Time, or SIBT) at and departure (Scheduled Off-Blocks Time, or SOBT) from the airport's apron — the designated area where it is stationed and maintained (Ashford et al. 2012). These activities necessitate the convergence of various resources such as staff and technical assets, including vehicles, around the parked aircraft to undertake all operations needed to prepare it for its next departure. The turnaround process varies based on the specific aircraft model/make, airline, ground handling service provider, etc. Symbiosis of the involved resources in supporting coordinated delivery across the many aircraft being turned around at any one time at the same airport is crucial. Disruptions of some activities often propagate quickly, causing undesired knock-on delays to other activities downstream. Maintaining the punctuality of operations and delays to an acceptable level is just as crucial.

4.2 Example Model Reuse

In his recent MSc dissertation (2023), one of the present authors (Almuzaki, the *modeler* in the following) studied ways to enhance the effectiveness of aircraft turnaround processes, by investigating various mixes of real-time operational management rules and their capacity to coordinate turnaround operations across the multiple service providers that normally operate such services at the same airport.

Almuzaki approached this project as a postgraduate taught student in Business Analytics, with prior experience of DES (including as a course tutor at universities), and tools such as Arena, but very little knowledge of ABS, SD, or Hybrid Simulation (HS), and tools such as Anylogic. Most importantly, he had no working experience in either airports or airlines, nor substantial knowledge of the field other than that from being a passenger. The learning curve for anyone wanting to work on the modeling and analysis of aviation operational problems but without a prior understanding of its intricate jargon is very steep indeed. The modeler can learn a great deal from the general literature on the subject (Ashford et al. 2012), but this takes time, and MSc dissertations in the UK are often about three months long. Close contact with project stakeholders from the industry may help to cut corners, but Almuzaki's was an academic project with no direct involvement from the industry. Access to other 'subject matter experts' (SMEs) was crucial though. In this case, two of the co-authors (Tomasella and Padrón) have fifteen years each of experience in working with airports and airlines in both research and consultancy capacities, whilst another co-author (Song) is currently reading for a PhD in simulation model reuse, and an expert in HS and tools such as Anylogic. Completing the project team were Ouenniche and Tomasella as research supervisors of both Song and Almuzaki.

Naturally, the modeler began with a focused effort on 'finding out'. In these earlier days, he drew rich pictures such as the one in Figure 2, with increasing sophistication as he kept learning more about the problem situation of his study. This in turn helped him to understand somewhat more *obscure* angles (at

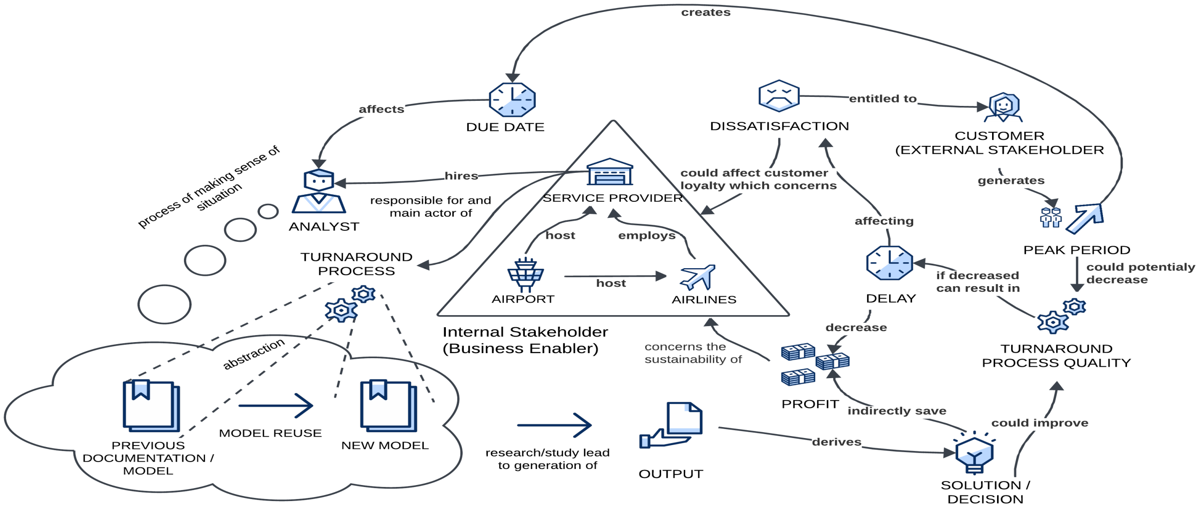


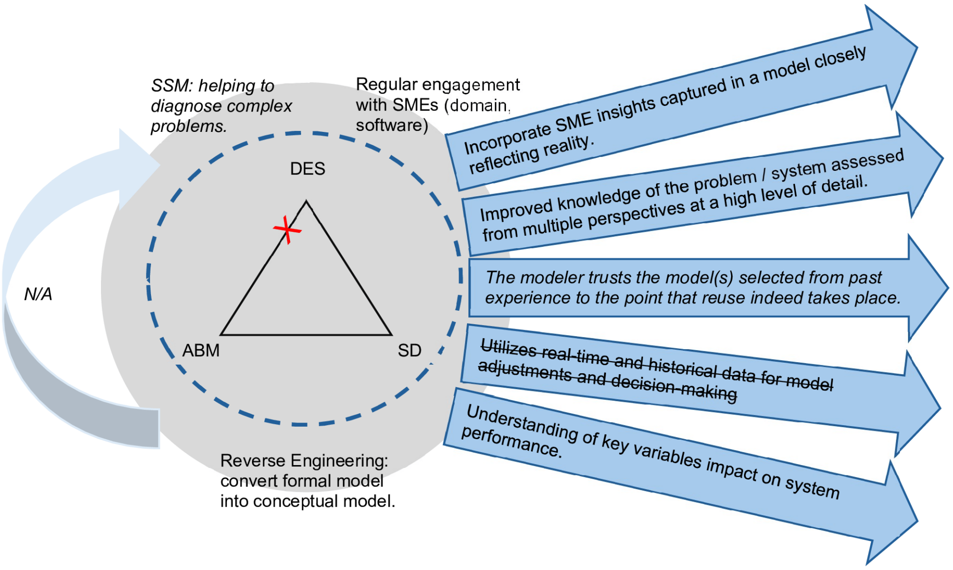
Figure 2: One of the earliest rich pictures produced as part of Almuzaki's work.

least from a non-expert such as him) from the documentation available on the previous projects carried out by the team in the same area and for which models were indeed available for reuse.

As the rich picture converged towards a stable version, the modeler started developing his own $S^{M}_{c,t}(AS-IS)$, which was produced, over a number of iterations, in the form of the 'list-of-six'. With the full draft of his conceptual model available in textual form, the modeler started harvesting across the field of previous projects available for reuse. Beyond the general information overload, he noticed that none of the otherwise extensively documented previous projects provided conceptual models (especially their graphical versions) that could be considered 'complete'. This may have different explanations. For instance, graphical conceptual modeling is often pursued as a sketch rather than as a detailed blueprint for the 'code to-be' (let alone as a programming language)—see our short discussion on this aspect in Section 2. Whatever the reason(s) behind it, it posed challenges to a relatively novice modeler, who had to quickly learn about various aspects of the project of which he had had no direct prior experience: (i) the problem domain itself, with its nuances; (ii) how to effectively and efficiently reuse related models developed by other people in different contexts; (iii) the various levels of hybridization that the previous works had developed to study their own versions of the problem; (iv) lack of prior exposure to Anylogic—retained as the software of choice. Without resolving (i–iv), the choice of what model(s) to reuse seemed an impossible one to make.

To resolve the issue (ii), the modeler was presented with the decision-making process in Figure 1. His immediate reaction was that all the elements in the process seemed pure common sense. At the same time, the modeler realized that, as a relatively inexperienced simulation modeler, he wouldn't have been able to draft a process reaching a similar level of detail to guide his study.

Once given the picture, he quickly developed an understanding of where modeling was involved, where validation was, and where some of the techniques and tools known to him could be adopted. One such *tool* was the framework (referred to by the authors as the 'kettle' or 'teapot') developed in Jones et al. (2022) to represent the *modeling frame*, which is "the choice and combination of modelling approaches, in hybrid modelling studies". This would quickly become crucial to help him resolve the issue (iii). In just a few days, the modeler used the teapot framework to document retrospectively the various aspects of hybridization from past projects. These aspects are: 1) the extent to which DES, ABS, and SD had been hybridized; 2) any hybridization of simulation with analytic models (hybrid modeling); 3) the hybrid modeling environment adopted (including methodological choices); 4) any hybridization related to the experimental approach adopted (e.g. with metaheuristic search); and 5) study outputs of various nature (e.g. beyond statistical analysis).



Component	Details
1) HS	· DES to model the sequence and timing of operations within the airport. · ABS to capture the behavior and interactions of individual agents (ground handlers, service vehicles). · *More detailed DES model of the sequential flow of turnaround operations.*
2) HS & analytic model	· N/A
3) Hybrid modeling environment	· *Use SSM to structure complex problems and engage SMEs in the modeling process.* · The conceptual model was presented with a highly descriptive structure that facilitated SMEs engagement, to ensure understanding of the modeling process and to create buy-in. · Incorporating SMEs engagement for generating a conceptual model, through reverse engineering. (~~Re-coding of the old model on a new software platform ensures the confidence of the model through corrective experiments.)~~
4) Hybrid experimental approach	· N/A
5) Hybrid outputs	· Engages different SMEs, (domain, software) in the modeling process to improve accuracy. · A model that engages SMEs while maintaining manageable complexity fosters a multi-perspective understanding of the problem/system, enhancing the modeler's relevant knowledge. · *Through iterative validation of overlapping components (in collaboration with SMEs), the modeler builds trust in the reuse of the selected model.* · ~~The selected models were recorded on the new software platform and the recoded models were calibrated with historical data to serve as benchmark models for reuse.~~ · Clear understanding of the impact of key input variables on outputs.

Figure 3: Modeling frame for Almuzaki's project.

Adoption of the teapot framework resulted in the choice to focus solely on Cattaneo (2018) and the related documentation and models, for three reasons. First, all the other projects —with the exception of Saggar et al. (2021)—featured a level of sophistication that the modeler didn't need, in that their hybridized experimental approach and hybrid modeling nature included deep integration with metaheuristics, as well as constraint and mathematical programming. Second, Saggar et al. (2021) had focused on turnaround technical assets, whilst Cattaneo (2018) had also modeled turnaround human resources. Third, the documentation of the conceptual models in Cattaneo (2018) was amongst the most extensive available.

Figure 3 shows the resulting modeling frame for Almuzaki's work. 'Normal' text refers to details retained from Cattaneo (2018), ~~strikethrough~~ text denotes elements from Cattaneo (2018) which did not apply to Almuzaki's project, and text in *italics* refers to new details that apply only to Almuzaki's work.

Finally, to resolve issues (i) and (iv), the modeler proceeded as follows. First, the Anylogic model components from Cattaneo's work were reverse engineered into a collection of network-like diagrams, one of which (the general model of an aircraft turnaround) is shown in Figure 4. These diagrams didn't follow any particular graphical notation or standard. They were built by the modeler to map his own understanding of the Anylogic implementation in Cattaneo (2018). They were then used by him during a series of meetings with Song (the 'Anylogic SME') to double-check the modeler's understanding of the

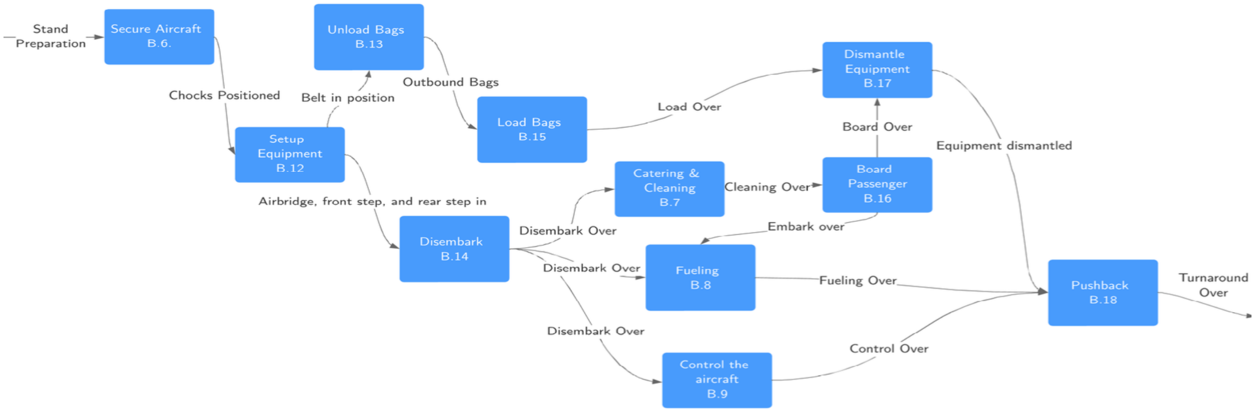


Figure 4: Part of conceptual model: perform turnaround structure flowchart.

computer code and, ultimately, the software tool, the learning of which, due to time pressure, was pursued by working directly on this complex, very realistic model. Therefore, in Almuzaki's experience, reverse engineering was not dictated by the lack of conceptual models. More crucially, the modeler had no prior understanding of how the formal model had been structured by Cattaneo to model different components of his system of interest, nor of whether or it represented well enough the vast collection of BPMN and other diagrams constituting Cattaneo's conceptual model. The latter point meant no trust could be placed by the modeler in Cattaneo's outputs, and some form of validation/verification was felt necessary.

Once a satisfactory level of understanding of the software had been developed, successful verification took place between Cattaneo's formal and conceptual models. This started to instill trust in Cattaneo's modeling and coding efforts on the side of Almuzaki, who by this point had developed a good working knowledge of the software tool as well. However, the modeler's still substantial lack of understanding of quite a few details about aircraft turnaround processes as found in either the conceptual or the formal models suggested that a second round of one-to-one meetings with the aviation SMEs (Tomasella and Padròn) should be held. Perhaps not everything that had been modeled by Cattaneo was 'correct' when compared to real-world versions of the studied processes. Given the similar validating role that the same aviation SMEs had played years earlier in Cattaneo's project, it was probably unlikely that any *major* modeling errors had been made. However the modeler had not been part of that experience, and the process of building trust in the models from the past had not yet been completed. Further meetings were held between the modeler and the aviation SMEs, the purpose of which was to validate, in all details, the BPMN diagrams, state charts, UML class diagrams, etc., of Cattaneo's work, vs the knowledge of the field possessed by the two SMEs. The further understanding that these meetings contributed to developing for the group, made it possible to correct the conceptual diagrams first, and the computer code next. At this juncture, both the conceptual and the formal models from Cattaneo (2018) were finally in the position of being reused in Almuzaki's work.

We are now at the bottom of Stage 2, where the information available in both $S^M_{c,t}(AS-IS)$ (built by the modeler for the current study) and $S^M_{c,t}(AS-WAS)$ (also built by the modeler, as a result of the thorough validation and verification of Cattaneo's models) may start to be exploited for actual model reuse. Following on from the many examples of tabular tools for conceptual modeling that are discussed in the textbook by Robinson (2014), the modeler then developed and employed tools such as Tables 1 and 2. The former table captures the indication of what components from the textual conceptual model of the current study are indeed included in (or excluded from) the final model, whether the component exists in the model(s) about to be reused, and some accompanying rationale. Table 2 goes one step further, whilst increasing the level of detail by one. The table indicates what sub-components are included in what model component, as well as specific notes of what details what sub-components of either the conceptual or the formal models need to be modified and how together with some accompanying rationale.

Table 1: Overlap examination of conceptual model components for reusability assessment (partial content).

Component	Include / Exclude	Present in previous work	Justification
Chocks activity	Include	Yes	Important discrete event process within system.
Ground handling team's dispatching rule	Include	Yes	Important as it is the key finding of previous work.
Procedure for service providers to request truck turnaround process	Exclude	No	The procedure details are not well-known.Truck companies are modeled to send their trucks once there is a service provider available to assist an incoming aircraft not yet having trucks to aid the turnaround process.
…	…	…	…

Table 2: Revamping the components of the previous conceptual model (partial content).

No.	Component name	Sub-component	Previous Model		Conceptual Model Correction		Formal Model Correction	
			Conceptual Model	Formal Model	Predefined Subcomponent By modeler	Remark and Justification by SMEs	Proposed Ideas by modeler	Remark and Justification by SMEs
1	Board Passenger	Ambulift for PRM logic	Ambulift for PRM is used when there are no connection.	Ambulift is used regardless of the connection's number.	Ambulift is not needed when there are no connections.	We do not use the ambulift for the PRM if an airbridge is used.	Remove the need for an ambulift if an airbridge is used, but add boarding delays for PRM passengers.	The proposed idea is validated as it reflects the justification in conceptual model correction.
2		Coaches Release	Used coaches were not released and returned to the station.	After embarking, coaches are released, a 'free' signal is sent, then move into the coach's store.	Coaches are released once embarking is over and returned to the coach station.	The existing simulation model is correct, but the conceptual model should reflect that. The proposed correction is justified.		
3	Secure the Aircraft	Logic in triggering equipment setup process	Asset setup equipment can be performed once the chocks are set.	Follows the conceptual model.	Set up equipment after placing chocks, security rope, and GPU.	The existing model should work well, but the process may imply simplification as the rope is usually set after the chocks are in place.	Setup equipment can be performed once chocks, security rope, and GPU are set	No change should be made, following the remark in conceptual model correction.
4	…	…	…	…	…	…	…	…

5 CONCLUSION

This research captured the essential elements of the decision-making process that probably *many* modelers follow (perhaps unconsciously) when reusing simulation models, and structured them in the flowchart form presented as the decision-making process in Figure 1. We also provided empirical evidence of where selected existing knowledge, in the form of frameworks (Jones et al. 2022), methodologies (Tako and Kotiadis 2015) or tools (e.g. rich pictures, various tables), maps directly onto Figure 1. This responds to one of the points made in the discussion at the end of Jones et al. (2022): "Additional work is needed to understand how our representation method can [sic] extended or incorporated into other frameworks and how new alternatives can help improve hybridization planning at the beginning of a project and the benefits this brings in relation to model quality, stakeholders, project time-frame, etc.". In our experience, we found that the 'teapot' framework proposed by Jones and colleagues can play an important role in the earlier stages of model reuse (Stages 1 and 2), wherever different elements of hybridization are found in previous models available for reuse. We believe our five-stage process will benefit the less experienced modelers in structuring their model reuse efforts. At the same time, we hope our discussion will help to convince more modelers that initiating model reuse *earlier*—i.e. at the conceptual level—may bring about considerable learning to the modeler in particular, and that the documented evidence that it generates is likely to boost even further the opportunities of model reuse (and time/cost savings) in the future.

REFERENCES

Ashford, N. J., P. Coutu, and J. R. Beasley. 2012. *Airport Operations*. 3rd ed. New York: McGraw-Hill, Inc.

Balci, O., J. D. Arthur, and R. E. Nance. 2008. “Accomplishing Reuse with A Simulation Conceptual Model”. In *2008 Winter Simulation Conference*, 959–965 https://doi.org/10.1109/WSC.2008.4736162.

Balci, O., J. D. Arthur, and W. F. Ormsby. 2011. “Achieving Reusability and Composability with A Simulation Conceptual Model”. *Journal of Simulation* 5(3):157–165.

Balci, O., G. L. Ball, K. L. Morse, E. Page, M. D. Petty, A. Tolk *et al*. 2017. “Model Reuse, Composition, and Adaptation”. In *Research Challenges in Modeling and Simulation for Engineering Complex Systems*, edited by R. Fujimoto, C. Bock, W. Chen, E. Page, and J. H. Panchal, Simulation Foundations, Methods and Applications, 87–115. Cham: Springer International Publishing.

Balci, O. and W. F. Ormsby. 2007. “Conceptual Modelling for Designing Large-scale Simulations”. *Journal of Simulation* 1(3):175–186.

Brailsford, S. C., T. Eldabi, K. Martin, N. Mustafee, and A. F. Osorio. 2019. “Hybrid Simulation Modelling in Operational Research: A State-of-the-Art Review”. *European Journal of Operational Research* 278(3):721–737.

Cattaneo, G. 2018. *“Analysis of Collaborative Real-Time Decision Making Policies for the Airport Team Allocation Problem in Ground Handling Operations"*. Ph.D. thesis, Politecnico Di Milano School of Industrial Engineering.

Checkland, P. and J. Scholes. 1999. *Soft Systems Methodology in Action*. Hoboken: John Wiley & Sons, Inc.

Coad, P. 1992. “Object-Oriented Patterns”. *Communications of the ACM* 35(9):152–159.

Dahmann, J. 1997. “High Level Architecture for Simulation”. In *Proceedings First International Workshop on Distributed Interactive Simulation and Real Time Applications*. January 9th-10 th, Eilat, Israel, 9 – 14.

El Haouzi, H., A. Thomas, and J. F. Pétin. 2008. “Contribution to Reusability and Modularity of Manufacturing Systems Simulation Models: Application to Distributed Control Simulation within DFT Context”. *International Journal of Production Economics* 112(1):48–61.

Fishwick, P. and N. Mustafee. 2019. “Broadening Participation in Modelling”. In *2019 Winter Simulation Conference (WSC)*, 1316–1327 https://doi.org/10.1109/WSC40007.2019.9004830.

Forrester, J. W. 2013. *Industrial Dynamics*. Illustrated ed. Connecticut: Martino Fine Books.

Fowler, M. 2018. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. 3rd ed. Boston: Addison-Wesley Professional.

Guo, S., C. Deng, Y. Wen, H. Chen, Y. Chang, and J. Wang. 2024. “DS-Agent: Automated Data Science by Empowering Large Language Models with Case-Based Reasoning”. *arXiv preprint arXiv:* 2402.17453.

Gök, Y. S., S. Padrón, M. Tomasella, D. Guimarans, and C. Ozturk. 2023. “Constraint-Based Robust Planning and Scheduling of Airport Apron Operations through Simheuristics”. *Annals of Operations Research* 320(2):795–830.

Gök, Y. S., M. Tomasella, D. Guimarans, and C. Ozturk. 2020. “A Simheuristic Approach for Robust Scheduling of Airport Turnaround Teams”. In *2020 Winter Simulation Conference (WSC)*, 1336–1347 https://doi.org/10.1109/WSC48552.2020.9383947.

Harwood, S. 2021. “Introducing the VIPLAN Methodology (with VSM) for Handling Messy Situations – Nine Lessons”. *Systemic Practice and Action Research* 34(6):635–668.

Jones, W., K. Kotiadis, J. R. O’Hanley, and S. Robinson. 2022. “Aiding the Development of the Conceptual Model for Hybrid Simulation: Representing the Modelling Frame”. *Journal of the Operational Research Society* 73(12):2775–2793.

Kovács, G. L., S. Kopácsi, J. Nacsa, G. Haidegger, and P. Groumpos. 1999. “Application of Software Reuse and Object-Oriented Methodologies for the Modelling and Control of Manufacturing Systems”. *Computers in Industry* 39(3):177–189.

Liu, Y., L. Zhang, W. Zhang, and X. Hu. 2016. “An Overview of Simulation-Oriented Model Reuse”. In *16th Asia Simulation Conference and SCS Autumn Simulation Multi-Conference*. October 8th-7th, BeiJing, China, 48 – 56.

Mingers, J. and J. Rosenhead. 2004. “Problem Structuring Methods in Action”. *European Journal of Operational Research* 152(3):530–554.

Nance, R. 1999. “Distributed Simulation with Federated Models: Expectations, Realizations and Limitations”. In *1999 Winter Simulation Conference (WSC)*, 1026–1031 https://doi.org/10.1109/WSC.1999.816816.

Nance, R. E. 1994. “The Conical Methodology and the Evolution of Simulation Model Development”. *Annals of Operations Research* 53(1):1–45.

Oral, M. and O. Kettani. 1993. “The Facets of the Modeling and Validation Process in Operations Research”. *European Journal of Operational Research* 66(2):216–234.

Overstreet, C. M., R. E. Nance, and O. Balci. 2002. “Issues in Enhancing Model Reuse”. In *International Conference on Grand Challenges for Modelling and Simulation*. Auguest 25th-30th, Wadern, Germany, 27 – 31.

Pace, D. K. 2000. “Ideas About Simulation Conceptual Model Development”. *Johns Hopkins APL technical digest* 21(3):327–336.

Peng, G., H. Mao, H. Wang, and H. Zhang. 2016. “BOM-Based Design Knowledge Representation and Reasoning for Collaborative Product Development”. *Journal of Systems Science and Systems Engineering* 25(2):159–176.

Pidd, M. and A. Carvalho. 2006. “Simulation Software: Not the Same Yesterday, Today or Forever”. *Journal of Simulation* 1(1):7–20.

Reese, R. and D. L. Wyatt. 1987. “Software Reuse and Simulation”. In *1987 Winter Simulation Conference (WSC)*, 185–192 https://doi.org/10.1145/318371.318404.

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

Robinson, S. 2014. *Simulation: The Practice of Model Development and Use*. 2nd ed. London: Bloomsbury Publishing.

Robinson, S. 2015. “A Tutorial on Conceptual Modeling for Simulation”. In *2015 Winter Simulation Conference (WSC)*, 1820–1834 https://doi.org/10.1109/WSC.2015.7408298.

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

Robinson, S., R. Brooks, K. Kotiadis, and D.-J. V. D. Zee. 2010. *Conceptual Modeling for Discrete-Event Simulation*. Boca Raton: CRC Press.

Robinson, S., R. E. Nance, R. J. Paul, M. Pidd, and S. J. E. Taylor. 2004. “Simulation Model Reuse: Definitions, Benefits and Obstacles”. *Simulation Modelling Practice and Theory* 12(7):479–494.

Saggar, S., M. Tomasella, G. Cattaneo, and A. Matta. 2021. “Enhanced Operational Management of Airport Ground Support Equipment for Better Aircraft Turnaround Performance”. In *2021 Winter Simulation Conference (WSC)*, 1–12 https://doi.org/10.1109/WSC52266.2021.9715320.

Saha, S., M. Tomasella, G. Cattaneo, A. Matta, and S. Padrón. 2021. “On Static vs Dynamic (Switching of) Operational Policies in Aircraft Turnaround Team Allocation and Management”. In *2021 Winter Simulation Conference (WSC)*, 1–12 https://doi.org/10.1109/WSC52266.2021.9715316.

Sargent, R. G., D. R. Mensh, R. Nance, H. Sallin, and J. F. Heafner. 1986. “Issues in Simulation Model Integration, Reusability, and Adaptability (Panel Session)”. In *1986 Winter Simulation Conference (WSC)*, 511 https://doi.org/10.1145/318242.318790.

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.

Tomasella, M., A. Clare, Y. S. Gök, D. Guimarans, and C. Ozturk. 2019. “Sttar: A Simheuristics-Enabled Scheme for Multi-Stakeholder Coordination Of Aircraft Turnaround Operations”. In *2019 Winter Simulation Conference (WSC)*, 488–499 https://doi.org/10.1109/WSC40007.2019.9004787.

US Department of Defence, D. o. D. 1993. “Integration Definition for Information Modeling (IDEF1X)”. Technical Report FIPS PUB 184, National Institute of Standards and Technology (NIST).

US Department of Defence, D. o. D. 1996. “High Level Architecture Federation Development and Execution Process (FEDEP) Model. Version 1.0”. Technical report, US Department of Defence.

Wyatt, D. 1990. “A framework for Reusability Using Graph-Based Models”. In *1990 Winter Simulation Conference Proceedings (WSC)*, 472–476 https://doi.org/10.1109/WSC.1990.129562.

Yilmaz, L. and T. I. Ören. 2004. “A Conceptual Model for Reusable Simulations Within a Model-Simulator-Context Framework”. In *Proceedings of CMS 2004 - Conference on Conceptual Modeling and Simulation*. October 28th-30th, Genoa, Italy.

AUTHOR BIOGRAPHIES

XIAOTING SONG is, at the moment of writing, a PhD candidate with the Management Science and Business Economics Group at the University of Edinburgh Business School. His research interests are in the theory and applications of model reuse. His e-mail address is Chenyi.Song@ed.ac.uk.

MAURIZIO TOMASELLA is a Reader at the University of Edinburgh Business School (UEBS). Previously at the University of Cambridge and Politecnico di Milano, Maurizio’s work employs Simulation-Optimization and Multi-Criteria Decision Analysis techniques to address applications in airport operations management. His e-mail address is maurizio.tomasella@ed.ac.uk.

LAZUARDI ALMUZAKI graduated in 2023 with an MSc in Business Analytics from the UEBS. His dissertation focused on the application of model reuse and hybrid simulation. His e-mail address is lazuardy.almuzaki@gmail.com.

JAMAL OUENNICHE is a Professor and Chair in Business Analytics at the (UEBS). Jamal’s contributions span a variety of areas including optimization, artificial intelligence, data envelopment analysis, multicriteria decision-making, performance evaluation and benchmarking, risk modeling and analysis, and forecasting. His e-mail address is Jamal.Ouenniche@ed.ac.uk.

SILVIA PADRÓN is an Associate Professor at TBS Business School (Toulouse, France). Her research is focused on metaheuristics and the hybridization of optimization methods with simulation techniques to solve stochastic and complex combinatorial problems in application areas such as transportation, aviation, and supply chain management. Her e-mail address is s.padron@tbs-education.fr.