Proceedings of the ASME 2018 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2018
August 26–29, 2018, Quebec City, Quebec, Canada

DETC2018-85317

OPERATIONAL AND STRATEGIC DECISIONS IN ENGINEERING DESIGN GAMES

Paul T. Grogan
School of Systems and Enterprises
Stevens Institute of Technology
Hoboken, New Jersey 07030
Email: pgrogan@stevens.edu

Alparslan Emrah Bayrak
Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
Email: bayrak@cmu.edu

ABSTRACT

Engineering design games model decision-making activities by incorporating human participants in an entertaining platform. This article distinguishes between design decisions at operational and strategic timescales as important features of engineering design games. Operational decisions consider static and short-term dynamic decisions to establish a player's situation awareness and initial entertainment. Strategic decisions consider longer-term dynamic decisions subject to large uncertainties to retain player engagement. However, constraints on cognitive load limit the ability to simultaneously address both lower-level operational design decisions and higher-level strategic decisions such as collaboration or sustainability. Partial automation can be introduced to reduce cognitive load for operational decisions and focus additional effort on strategic decisions. To illustrate tradeoffs between operational and strategic decisions, this paper discusses example cases for two existing games: Orbital Federates and EcoRacer. Discussion highlights the role of automation and entertainment in engaging human participants in engineering design games and makes recommendations for design of future engineering design games.

1 INTRODUCTION

Research in engineering design increasingly considers a broad class of real-world systems going beyond traditional boundaries of engineered artifacts to interact with and contribute to key societal functions [1]. In addition to the short-term, well-bounded *operational* decisions, design of engineering systems requires careful consideration of their large scale, long lifetime, and complex features such as adaptation, self-organization, and emergence [2]. Pursuing objectives such as sustainability requires an integrated perspective to understand interdependencies at multiple levels of abstraction [3]. These types of *strategic* decisions make significant resource commitments with often-irreversible investments and a broad scope of intended impact [4] and demand a different class of analysis methods than operational decisions more common to engineering design.

Human intuition is a useful resource to leverage when addressing the complexity of engineering system design. Games and interactive simulations are natural models of human decision-making, interaction, and feedback with a long history of applications in military, policy, educational, and business activities [5] and, more recently, in engineering design [6]. Abt describes games as "an activity among two or more independent decision-makers seeking to achieve their objectives in some limiting context" [7] which, slightly extended, also considers individuals interacting with an unknown or uncertain environment. This characterization broadly applies to design activities, whether decision-makers represent disciplinary experts in a single organization or multiple firms with conflicting objectives.

The human participant represents a significant opportunity to incorporate social factors into a design problem but also a critical challenge to overcome limited cognitive load, high costs, and long timescales. Focusing a game on lower-level operational decisions consumes significant time and cognitive load, limiting the ability to address higher-level strategic issues. Meanwhile, focusing on higher-level strategic decisions by automating other decisions omits technical details and may fail to initially engage human participants into the problem. A careful balance of automation allows participants to tackle strategic issues while maintaining situation awareness of technical design.

To facilitate the study of engineering design problems using games, this paper distinguishes between decisions at operational and strategic timescales, followed by specific recommendations for their representation in games. First, this paper contributes a model of engineering decision-making as a bi-level problem with operational decisions at the lower level and strategic decisions at the higher level. Next, model implications regarding automation and entertainment are discussed in the context of engineering design games. Two example cases describe how operational and strategic decisions fit within design games previously developed by the authors. Finally, a discussion draws comparisons across the two cases and summarizes key conclusions.

2 ENGINEERING DECISION-MAKING

Outputs of design activities describe technical *artifacts* (e.g. constructs, models, methods, and instantiations) to be evaluated for *utility* or *value* provided to users [8]. Rational design decision-makers select the expected value-maximizing artifact alternatives; however, large design spaces and uncertain mapping between design decisions and value complicate the process [9]. While design can broadly be defined as a decision-making process, this section distinguishes between decisions for the physical system (static design decisions) and decisions for its behavior in time (dynamic decisions) at operational and strategic timescales.

2.1 Static Design Decisions

Static design decisions reflect the most common notion of design as a structural artifact description. In particular, the system architecture maps stated or desired features in the functional domain to elements of form or structure in the physical domain.

Design decision-making relates functional and physical domains to assess utility or value provided to users. For example, Axiomatic Design Theory (ADT) [10] models a system architecture as a vector $y = [y_1, \dots, y_M]$ of M functional requirements (FRs) and a vector $x = [x_1, \dots, x_N]$ of N design parameters (DPs), also referred to as design variables, related by design model $\mathbf{M}$ in Eqn. (1).

$$y = \mathbf{M}(x) \tag{1}$$

DPs can be interpreted as individual design decisions and FRs as resulting attributes to be compared against desired requirements. The design model $\mathbf{M}$ embodies physical and natural laws which relate physical form (DPs) to function (FRs). A simple linear design model may represent $\mathbf{M}$ as the vector product of x and a $M \times N$ matrix with elements $m_{ij} \approx \partial y_i / \partial x_j$.

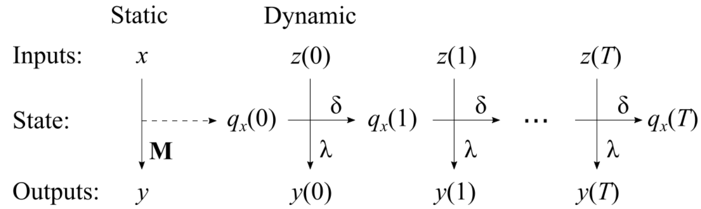

FIGURE 1. MODEL OF STATIC AND DYNAMIC DECISIONS

Value-centric or value-driven design processes rely on a system value model to map a set of DPs to a single scalar FR $V(x) \in y(x)$ expressing decision-maker preference or value [11]. From this perspective, design decisions aim to solve the optimization problem in Eqn. (2) to find the value-maximizing design alternative $x^\star$ from the set of candidates in design space $\mathbf{X}$.

$$x^\star = \arg\max_{x \in \mathbf{X}} V(x) \tag{2}$$

While it is desirable to quantify the overall value of a design x from a static perspective, most measures of performance depend also on dynamic behavior addressed in the following section.

2.2 Dynamic Design Decisions

Dynamic decisions govern the behavior of a system over time and are critical to evaluate design decisions based on anticipated performance. A decision rule or policy $z \in \mathbf{Z}$ prescribes specific temporal behaviors $z(t)$ in response to stimuli at time t. Relating dynamic decisions to static design decisions requires elaboration of a discrete-time dynamic system formalization based on [12]. Fig. 1 illustrates a model with such formalization by defining the model state q_x, state transition function δ, and output function λ.

The model state $q_x(t)$ captures all information relevant to the static design x as a function of time t starting from initial conditions, i.e. $q_x(0)$. The model state represents a snapshot in time including the complete system configuration, accumulated resources including key performance measures, and contextual features of the environment.

A state transition function δ in Eqn. (3) propagates the model state in time as a function of the current state $q_x(t)$ and behavioral inputs $z(t)$.

$$q_x(t+1) = \delta(q_x(t), z(t)) \tag{3}$$

The state transition function describes how a behavior influences or changes model state including how system functions change internal or external state variables, how to tabulate key performance measures, and how contextual uncertainty resolves over time.

A Mealy-type output function λ in Eqn. (4) returns temporal system attributes $y(t)$ as a function of the model state $q_x(t)$ and behavioral inputs $z(t)$.

$$y(t) = \lambda(q_x(t), z(t)) \tag{4}$$

The output function describes how the model state and behaviors relate to derived quantities of interest, allowing only fundamental or elementary information to be maintained as model state.

To reinforce these concepts, consider concepts from the infrastructure system-of-systems (ISoS) framework which helps define behavioral rules for infrastructure models [13]. State variables record key information such as resource contents and spatial locations of infrastructure elements. Behavioral functions modify state variables by adding (generating) or removing (consuming) resources from contents or transporting an element to change its spatial location. Key performance metrics such as net present value are based on time-discounted financial resources.

A revised view of design includes both the static design architecture and the dynamic decision policy expressed as a 2-tuple $(x,z) \in \mathbf{X} \times \mathbf{Z}$. Finding a decision policy z for design x resembles an optimal control problem to maximize a scalar measure of preference or value $V_{x,z}(t) \in y(t)$ evaluated at end-of-life time T in Eqn. (5).

$$V(x) = \max_{z \in \mathbf{Z}} V_{x,z}(T) \tag{5}$$

Note this expression abstracts time from a design decision, converting the dynamic decision into a static one.

2.3 Bi-level Model of Engineering Decision-making

This section develops a bi-level model of engineering design decision-making to repartition dynamic decisions to *operational* and *strategic* timescales such that $z = (o,s) \in \mathbf{O}_s \times \mathbf{S}$. On opposite ends of a spectrum, strategic decisions involve significant resource commitments and changes to the scope of an engineering system while operational decisions do not [4]. Strategic decisions effectively constrain the set of available operational decisions $\mathbf{O}_s$ illustrated in Fig. 2. The distinction between operational and strategic timescales is similar to past literature identifying epochs as temporal regions with a fixed context and eras as sequences thereof [14]. Despite these descriptions, some judgment is required to designate a dynamic decision as either operational or strategic and is dependent on the specific design problem.

2.3.1 Operational Design Decisions.

Dynamic decisions on an operational timescale follow a strategy-specific policy $o \in \mathbf{O}_s$ to determine how to structure and use a design's functional properties to achieve near-term objectives. For example, operational functions within the ISoS framework include storing, transforming, and transporting resources and transporting elements between locations subject to nominal efficiency factors.

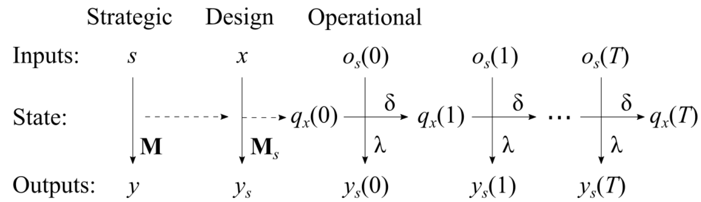

FIGURE 2. BI-LEVEL MODEL TO REPARTITION DECISIONS

Operational design decisions evaluate system value in Eqn. (6) assuming a fixed strategy $s \in \mathbf{S}$ as a well-defined context such as a nominal operational regime or mode of interaction with other decision-makers.

$$V_s(x) = \max_{o \in \mathbf{O}_s} V_{x,(o,s)}(T) \tag{6}$$

Thus, operational design decisions can be framed as a search for the best strategy-specific design $x_s^\star \in \mathbf{X}$ (and dependent operational policy) in Eqn. (7) as a slight refinement to the previous static formulation in Eqn. (2).

$$x_s^\star = \arg\max_{x \in \mathbf{X}} V_s(x) \tag{7}$$

Operational design decisions incorporate idealized dynamic decisions but are inherently constrained to a specific strategy.

2.3.2 Strategic Design Decisions

Dynamic decisions on a strategic timescale conforming to a strategic policy $s \in \mathbf{S}$ determine how to modify a system's architecture to achieve long-term objectives. At the strategic level, most quantities including contextual or environmental factors and interactions with other decision-making agents are highly uncertain. Strategic decisions are related to the design's non-functional properties ("ilities") such as flexibility, adaptability, and resilience [15]. For example, strategic functions within the ISoS framework include transforming elements to commission, decommission, or reconfigure operations and exchanging resources with other decision-making agents governed by social contracts.

Strategic design decisions evaluate system value in Eqn. (8) to compose results of operational design decisions.

$$V(s) = \max_{x \in \mathbf{X}} V_s(x) = V_s(x_s^\star) \tag{8}$$

Analogously, strategic design decisions can be framed as a search for the best strategy $s^\star \in \mathbf{S}$ in Eqn. (9).

$$s^\star = \arg\max_{s \in \mathbf{S}} V(s) \tag{9}$$

From this perspective, strategic design decisions take place on a higher level of abstraction assuming lower-level design and operational decisions have already been addressed. Note that some static design decisions that can change the architecture of the design problem are also part of the strategic decisions.

Distinguishing between two levels of decision-making helps to frame efforts to study design decision-making. In particular, studying strategic design decision-making in Eqn. (9) requires abstraction or automation of operational decisions which, in turn, reduces feedback available to the design decision-maker from operational design activities in Eqn. (7). The desired balance between operational and strategic decisions depends on several factors discussed in the next section.

3 ENGINEERING DECISION-MAKING IN GAMES

Games present a natural medium to study engineering decision-making by placing human players in a synthetic environment to engage with and receive rapid feedback from engineering problems. Digital and online games, in particular, provide an opportunity to collect large quantities of information by tapping into a global community of players.

Developers of engineering design games must carefully balance operational and strategic decision-making to achieve game objectives. Fundamental differences make it difficult to achieve depth at all levels simultaneously and there are also challenges to considering strategic decision-making in isolation. Cognitive limits of players, game elements and mechanisms to foster engagement must be considered along with the goal of the engineering design research when developing games for that purpose. This section discusses automation and entertainment as two important dimensions of engineering game design.

3.1 Automation of Design Decisions

Interaction with human players simultaneously represents an enormous opportunity and significant challenge to employing engineering design games. Humans use games as a new form of information feedback within a synthetic environment to learn about complex systems [16]. However, human decision-making is slow, expensive, and limited by high cognitive loads in unfamiliar environments representing information at different levels of abstraction than those developed through expertise [17].

Studies of design as a decision-making activity require consideration of both static and dynamic decisions. Analysis in stable contexts may only depend on operational decisions; however, broader studies of strategic issues can be burdened by operational decisions. Automating operational decisions using optimization methods such as linear programming (LP), integer programming (IP), and dynamic programming (DP) can facilitate strategic analysis by increasing the level of abstraction for human input. Resulting strategic analyses benefit from a wider class of methods including scenario planning, real options, and game theory to reason across highly uncertain contexts.

Automating operational decisions also carries downsides. For example, the U.S. defense wargaming community expended considerable efforts in the 1980s to automate player decisions and permit multi-scenario analysis of strategic issues [18]. However, later reports call to re-insert humans into a family of strategic analysis tools, recognizing analytic techniques alone cannot address strategic uncertainties and human participation spurs innovation and improves feedback to real decision-makers [19]. From this perspective, participating in operational decisions gives players situation awareness to understand the context of more abstract strategic decisions [20].

Strategic design games must carefully balance automation to abstract tedious or well-understood behaviors (the characterization of which is subjective to each participant) while maintaining situation awareness for players.

3.2 Entertainment in Engineering Design Games

Entertainment is an essential game element to engage players and represents a technique to facilitate data collection in engineering design. Access to a large community of self-motivated players presents significant opportunities to advance research by collecting large data volumes and creative solutions. However, designing games that are both useful for research and entertaining for participants presents a challenge as the two objectives often impose conflicting objectives.

A body of literature addresses important game design elements for entertainment purposes. For example, several principles to improve engagement include understanding the player and social elements, user interface, game theme and characters, and game mechanics [21]. However, few works translate entertainment game design principles to more specific classes including engineering design games. These more meaningful applications require a trade-off between triadic game elements of play, meaning, and reality [22]. Play is most closely related to entertainment but is linked to a freedom to fail in a game's virtual world. Meaning refers to the mechanics that foster learning and intuition during game play, linking actions within a game to research objectives set outside. Reality refers to the actual engineering problem of interest, establishing literal correlation between games and the reality they model. Game design decisions frequently create conflicts among these elements and a balance must be sought for successful implementations [22].

Focusing only on players' decisions as a subset of all game

FIGURE 3. ORBITAL FEDERATES GAME BOARD

design features, operational and strategic decisions should be considered separately as they serve different entertainment purposes. While operational design decisions help players gain familiarity with the game in the short term, strategic decisions provide multiple contexts to these decisions and maintain long-term interest. Games completely lacking operational decisions may suffer from abstraction and fail to initially engage players and games lacking strategic decisions may suffer from repetition and fail to maintain player engagement.

3.3 Examples from Existing Games

To illustrate the balance between automating decisions while maintaining player entertainment and engagement, the following sections provide examples of two existing design games created by the authors. Each case describes the static and dynamic decisions implemented in the games, discusses how automation could be incorporated to abstract operational decisions, and what strategic analysis could be enabled. Subsequent discussion summarizes the key insights from both games, and synthesizes key contributions to inform both development and use of engineering games to study design decision-making.

4 EXAMPLE CASE: ORBITAL FEDERATES

Orbital Federates is a multi-player engineering design game to study strategic issues of collaboration in federated satellite systems (FSS) [23]. Originally developed as a tabletop board game illustrated in Fig. 3, later work implemented an automated Python simulation to analyze strategic design decisions [24].

Players in Orbital Federates design and operate simplified models of space systems to accumulate revenue by completing data contracts from third parties. Randomly-generated contracts demand data from spatially- and temporally-specific observations of phenomena to be down-linked to ground stations. Players use their own systems or collaborate with others to complete contracts. Expected net present value (time-discounted revenue less costs) is a clear measure of design value in Orbital Federates.

4.1 Operational Design Decisions

Static design decisions determine the overall FSS structure including surface-based ground stations and space-based satellites. Ground stations are located at one of six radial sectors partitioning the Earth's surface. Satellites are located above an initial sector at an orbital altitude determining temporal propagation. Low orbits have the shortest period (three turns per orbit) than medium orbits (six turns per orbit) and stationary orbits remain fixed above one location.

Ground station and satellite designs specify a set of hardware modules which provide the following functions:

1. **Sensor**: generate and store up to one data unit specific to an observed phenomenon (e.g. visual or infrared spectra).
2. **Storage**: store up to one additional data unit.
3. **Inter-satellite link** (ISL): transmit and receive up to one data unit per turn between two satellites.
4. **Space-to-ground link** (SGL): transmit and receive up to one data unit per turn between satellites and ground stations.
5. **Shielding**: protect a satellite from damaging events.

Links designate either a proprietary or open protocol to restrict communication within one player's systems or allow potential services between players. Ground stations allow up to three modules while satellite buses allow up to two (small), four (medium), or six (large). System designs incur a fixed cost to cover essential platforms, hardware modules, and launch (for satellites).

Dynamic decisions at an operational timescale allow players to execute FSS functions enabled by hardware modules each turn. Key operational decisions include the following:

1. **Contracting** a demand by agreeing to deliver its requested data within a fixed time window.
2. **Sensing** a phenomenon to generate data for a contract.
3. **Storing** or **retrieving** data in a sensor or data storage unit.
4. **Transmitting** or **receiving** data between two links for relay (ISL) or down-link (SGL).
5. **Exchanging** financial resources between players for FSS data services such as relay, down-link, or storage.
6. **Resolving** a contract by delivering data to a ground station.

Tabletop game play uses tokens to support operational decision-making by denoting data contents and remaining link capacity. While players generally search for efficient paths to deliver data, they must also balance the uncertain availability of future contracts and manage scarce resources such as down-link capacity.

4.2 Strategic Design Decisions

Strategic decisions in Orbital Federates emphasize two dimensions. First, players follow a capacity expansion strategy to scale up space and ground systems throughout a game. Using revenue collected from completed contracts, players can purchase and operate new systems to augment existing capacity. While existing ground stations can be modified, for example to add a second link to increase bandwidth, existing satellites cannot be changed. Alternative strategies may lead players to pursue few large and expensive monolithic satellites or more numerous smaller and inexpensive distributed satellites.

Second, players follow a collective action strategy to guide whether and how to interact with other players, for example, to purchase and sell relay and down-link services. Using links with open protocols requires significantly more upfront capital than proprietary alternatives and is only beneficial if other players also participate as a federation. Once required functionality is in place, players must also agree on mechanisms to distribute financial resources in return for services.

4.3 Automating Operational Decisions

Operational decisions contribute high cognitive load in Orbital Federates, especially for large FSS, due to large dimensionality of the operational decision space $\mathbf{O}_s$. Efficient routing of resources through a network is well-characterized by a mixed-integer linear programming (MILP) optimization model implemented in the Python variant of Orbital Federates. The model is formulated and sequentially solved at each time step to determine operational decisions.

The MILP relies on a time-expanded network to maximize value of resolved contracts over a short-term timescale (typically six turns). It is characterized by a network flow glyph in Fig. 4 for system i illustrated over two time steps. Binary decision variables model data operations for notional contract c including z_{ic}^{sense} to sense data, $z_{kipc}^{\text{transport}}$ to receive data from system k using protocol p, $z_{ijpc}^{\text{transport}}$ to send data to system j using protocol p, z_{ic}^{store} to store data, and z_{ic}^{resolve} to resolve data. Integer constraints restrict functions based on storage and link capacity, spatial link availability, logical requirements (e.g. only satellites can sense data, only ground stations can resolve data), establish boundary conditions including existing data E_{ic}, and maintain net data flow conditions.

The objective function maximizes net revenue from completed contracts over a six-turn horizon. While stochastic programming could accommodate uncertain future demand availability, this deterministic MILP assigns a penalty for storing data as an opportunity cost. Several variants of the MILP operations model characterize strategies including independent operations by a single player, centralized operations to maximize total net revenue across multiple players, and federated operations to maximize individual revenue subject to a fixed price for opportunistic ISL and SGL services between players.

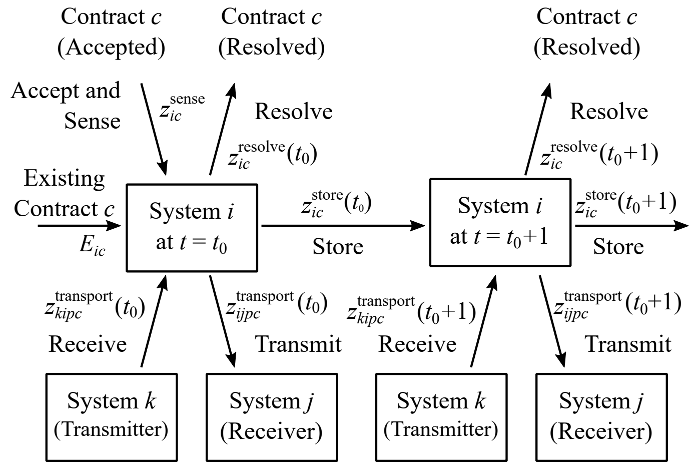

FIGURE 4. TIME-EXPANDED NETWORK GLYPH

TABLE 1. STRATEGIC DESIGN GAME FOR $n = 2$ PLAYERS

Player 1 Strategy (s_1)	Player 2 Strategy (s_2): Independent (ϕ)	Player 2 Strategy (s_2): Federated (ψ)
Independent (ϕ)	$V_1(\phi,\phi)$, $V_2(\phi,\phi)$	$V_1(\phi,\psi)$, $V_2(\phi,\psi)$
Federated (ψ)	$V_1(\psi,\phi)$, $V_2(\psi,\phi)$	$V_1(\psi,\psi)$, $V_2(\psi,\psi)$

4.4 Analyzing Strategic Decisions

No existing work addresses capacity expansion strategy in Orbital Federates. Instead, strategic analysis in [24] focuses on collective strategies for $n = 2$ players. Each player chooses between independent operations using proprietary link protocols (labeled as strategy ϕ) and federation operations using open link protocols for opportunistic fixed-price services (labeled as strategy ψ). The value function from Eqn. (8) takes on a multi-actor vector form in Eqn. (10) to accommodate the 2-tuple of strategy decisions $s = (s_1, s_2)$ and resulting actor-specific value.

$$V(s) = V(s_1, s_2) = \begin{bmatrix} V_1(s_1, s_2) \\ V_2(s_1, s_2) \end{bmatrix} \tag{10}$$

Strategy decisions are selected from the binary alternatives $s_1, s_2 \in \mathbf{S} = \{\phi, \psi\}$ and the resulting strategic design game in Table 1 expresses the strategic decision as a normal form game-theoretic problem.

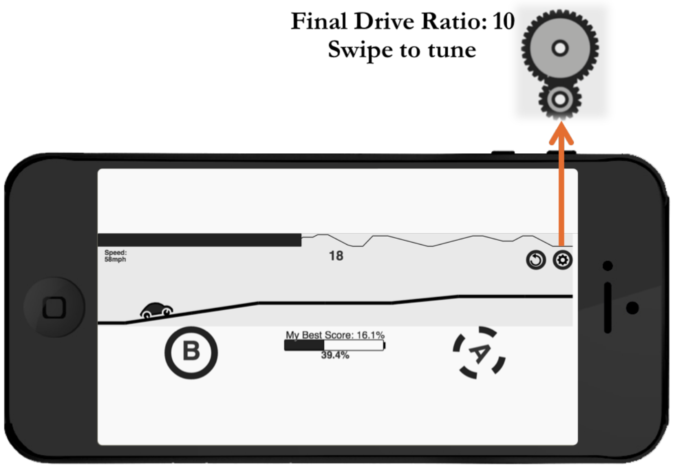

FIGURE 5. ECORACER USER INTERFACE

Within the fixed context of each strategy pair (s_1, s_2), players follow guidelines in Eqn. (7) to design and operate value-maximizing systems subject to potential interactive effects. Selection of an ideal strategy is no longer as simple as previously expressed in Eqn. (9) because one actor's value depends on the actions of the other. Resulting strategic analysis identifies stable strategy sets (e.g. Nash equilibria) or helps to characterize risk or payoff dominance of multiple equilibria [24].

5 EXAMPLE CASE: ECORACER

EcoRacer shown in Figure 5 is an online competition-based electric vehicle design and control game [25]. This game has been developed to compare human problem solving skills with a computational method and study ways to combine these two to improve solution quality [26]. Here, we analyze this game in the context of the bi-level decision-making framework presented in this paper.

The main goal in the game is to find the best powertrain design and the corresponding vehicle control strategy to complete a deterministic race track within a fixed time limit using minimum battery energy. Players are asked to play this game multiple times to improve their own score against other online players on a leader board. The original game contains static and dynamic design decisions at the operational timescale. While this game does not have decisions at the strategic timescale, the goal of this example is to discuss the outcomes of lacking such strategic design decisions. After describing the problem in the original game, this section discusses potential ways to improve the game play experience and use the same platform to ask a broader range of questions by adding an additional layer of strategic timescale.

5.1 Operational Design Decisions

The problem at an operational timescale in the original EcoRacer game involves a single static variable that corresponds to the final drive ratio in the powertrain. This variable is presented to users with a slider that can take a range of values. This game keeps the static design problem simple enough to make it easy to solve by both the human players and the computational methods for comparison. However, the problem can be extended to include other powertrain components parameters such as electric motor parameters or the battery size. The drive ratio static design decision affects the following performance metrics:

1. **Acceleration:** increasing the final drive ratio increases the torque multiplication at the vehicle output and affects the power delivered to the vehicle combined with the motor torque limits in a nonlinear way.
2. **Top speed:** increasing the final drive ratio increases the speed multiplication at the motor and linearly reduces the maximum vehicle speed due to the motor speed limits.
3. **Powertrain efficiency:** the final drive ratio affects the operating points of the motor and changes the powertrain efficiency due to nonlinear motor losses as a function of the motor speed and torque.

The first two performance metrics contribute to completing the race track quickly while the last metric helps achieve a high game score. Static design decisions alone do not allow players to evaluate their decisions and an operational problem must be solved to obtain a combined evaluation. Therefore, game players may make good static design decision but still receive a low score due to poor operational decisions.

Dynamic decisions at an operational timescale in EcoRacer determine vehicle control. The game provides two inputs to control the vehicle:

1. **Accelerate:** provides an increasing motor torque (up to the limits) to the powertrain and consumes battery energy based on the power and efficiency of the motor at the corresponding operating point.
2. **Brake:** provides a fixed negative torque to the powertrain to decelerate the vehicle and regenerates some energy based on the motor efficiency at the corresponding operating point.

Given the powertrain design set by the static decisions, players use these two inputs to control the vehicle fast enough to complete the track and efficiently enough to receive a high score. The game screen displays the speed of the vehicle, distance to the end of the track, and remaining battery energy in real time.

5.2 Automating Operational Decisions

Results in EcoRacer are sensitive to operational decisions. Also, as discussed in [25], a very small portion of the space of operational decisions can finish the track on time, making the game challenging for many players. To aid player engagement and facilitate a strategic decision-making process discussed in the next section, these decisions can be automated with existing methods in both control theory and machine learning.

Parametric vehicle design and control can be modeled as an optimization problem to minimize battery energy consumption integrated over time by varying the motor torque output as a time-dependent quantity and gear ratio as a static quantity. Static design problem can be formulated as a continuous nonlinear optimization problem and solved using existing methods. Control methods require a system of differential equations to model the relationships between system states and actions. Due to problem simplicity in EcoRacer, a model with a state variable corresponding to the vehicle position and an action variable for the motor torque is sufficient to capture the system dynamics. Including the constraints on the motor torque and speed, existing methods commonly used in vehicle control, such as dynamic programming [27] and Pontryagin's minimum principle [28] can be applied for automated decisions.

As an alternative to the control theory, machine learning relies on finding network relationships between a discretized representation of the variables (states and actions) as opposed to a system of differential equations. Methods such as reinforcement learning use game play data to train the parameters of a policy or a network governing the actions based on the system state [29]. Existing implementations of reinforcement learning can play classic Atari games [30]; however, these methods require a large amount of game play data created by testing the trained policy on the game platform.

5.3 Strategic Design Decisions

The dynamic design problem in EcoRacer currently contains only operational decisions due to the scope of the original work. As mentioned in [25] many of the players quit after a few plays, possibly because the game was difficult or non-intuitive. Adding decisions on a strategic timescale may put the operational decisions into new contexts and help improve the gameplay experience. Potential extensions with strategic decisions also lead to new research questions with a broader perspective.

One possible extension could be adding a fleet-level problem for a car sharing system from a business perspective. Prior work in [31] addresses the design of autonomous car sharing system design problem that includes static design decisions from vehicle powertrain design, charging system locations, and operational decisions from fleet assignment problems using system coordination and design optimization methods. These approaches might be intractable when the size of each subproblem increase or when strategic decisions are included. A game-based approach can help to leverage human intuition on complex system design problems where new strategic decisions including managing investments under future uncertainty, finding the optimal fleet mix, architectural design decisions for a variety of vehicles in the fleet can be analyzed. In that context, the operational decisions such fleet management can be used as tools to improve player engagement but optionally be automated using the research in design optimization to reduce the cognitive load.

Another layer of analysis could include a real-time competitive decision-making against an intelligent opponent. In the context of car sharing system, the same strategic decisions must be made in a different way when there is a common customer base with a competitor. Earlier work in vehicle design have studied competitive market systems [32, 33]. While methods such as game-theory can be used to model equilibrium scenarios under rational decision-making, online video games allow including human factors to model more realistic strategic decisions.

6 DISCUSSION

6.1 Key Insights from Orbital Federates

The original tabletop variant of Orbital Federates served as an effective medium to engage players but was too time-consuming to collect data for quantitative analysis of players' design decisions. The MILP optimization model facilitates analysis of collective strategies by automating operational data routing decisions; however, results have only been used within an analytic frame without human participation. The strategic design activity of choosing a collective strategy does not include any of the engaging activities of selecting static designs or operating them in a simulated space system. Key areas of future work may augment human interaction with a digital version of Orbital Federates to automate some operational decisions while retaining authority over strategic decisions including capacity expansion and collective action.

6.2 Key Insights from EcoRacer

EcoRacer provides an online platform that can potentially reach a large number of players. Operational decisions in EcoRacer provide a useful medium to study how human input can help the search with computational methods since the original problem can be formulated as an optimization problem with well defined objectives and design variables. On the other hand, the game suffers from maintaining long-term engagement for the majority of players due to lack of consideration of human factors in the game design. This outcome can partly be attributed to the lack of strategic decision-making in the game. The current implementation with only operational decisions provides a single context which penalizes failures severely. Adding strategic decisions can keep the players engaged longer by providing multiple contexts. In that case, failures at the operational timescale can be more tolerable which makes the game play more entertaining for new users. Also, a strategic timescale creates new research opportunities useful for a larger community of researchers. Considering the cognitive load of the players, a strategic design analysis can benefit from automating operational decisions using approaches from control theory and machine learning.

6.3 Cross-case Comparison

Orbital Federates and EcoRacer present two examples of engineering design games with different decision-making objectives but similar challenges regarding automation and entertainment. Differentiating between operational and strategic design decisions helps to identify which portions of game play may benefit from automation contributing to a balance between short-term and long-term entertainment.

Orbital Federates initially targeted strategic design issues but suffers from slow operational decisions and long design cycles. Automating portions would help to abstract operational decisions and allow players to focus on strategic issues while retaining entertainment. EcoRacer targets operational design decisions but suffers from repetitive game play and lack of long-term engagement. Automating portions of EcoRacer may reduce repetition but will also require adding strategic design issues to retain entertainment over longer periods.

7 CONCLUSION

This paper highlights fundamental differences between operational and strategic decisions in the context of engineering design and, specifically, games as interactive models. Both levels of decision-making are essential components to study engineering systems; however, they are not equal. Operational decisions execute functional behaviors to achieve short-term objectives under well-characterized conditions. Strategic decisions adapt the system architecture in response to contextual changes.

While strategic games may automate operational decisions to focus human effort on strategic design issues, there is good reason to consider some level of operational decision-making. Operational decision-making may improve short-term participant engagement by providing a "fun" activity. Over a longer term, operational decisions become more routine and strategic decision-making retains participant interest. Early automation of operational decisions may prevent participants from engaging in a design problem by losing sight of the operational context.

This paper leads to several interesting questions to be answered with further research using human subjects. This paper mainly discusses engineering design games from a technical perspective with a focus on classifying the types of decisions, but a discussion on human factors is left to a future study. Triadic game design highlights play, meaning, and reality as three important dimensions to consider when developing engineering design games. However, more prescriptive approaches on how to incorporate these three dimensions in different engineering system design problems are still missing. For instance, the amount of automation to incorporate into game design to balance cognitive load and entertainment is an important element to be addressed for future research. Also, determining how much variation exists among game players in terms of their preferences and how much of the problem context impacts the outcomes are important research directions for future work. Addressing these issues could help the researchers to develop games for their own purposes while achieving a good balance between strategic and operational timescales. Existing literature on entertainment game design and human factors including situation awareness could potentially influence this line of research.

Additional work is also required to formalize normative models of design as search or optimization problems as introduced in this paper. Differentiating between operational and strategic levels of design and incorporating multi-actor value functions adds complexity. In particular, combining computational methods with human input for operational and strategic decision-making in engineering design remains another open question for future research. While optimization methods can serve as a platform that can benefit from human inputs in the operational decision-making, the nature of strategic decisions makes it challenging and sometimes does not allow formulating optimization problems with well-defined objectives and design variables. A potential approach for future research could use human decision-making skills as a platform that can be supported by optimization methods.

ACKNOWLEDGMENT

This material is based in part upon work supported by the National Science Foundation under Grant No. CMMI–1742971.

REFERENCES

[1] O.L. de Weck, D. R., and Magee, C., 2011. *Engineering Systems: Meeting Human Needs in a Complex Technological World*. MIT Press, Cambridge, Massachusetts.
[2] Ottino, J., 2004. "Engineering complex systems". *Nature,* ***427***, p. 399.
[3] DeLaurentis, D., and Callaway, R., 2007. "A system-of-systems perspective for public policy decisions". *Review of Policy Research,* ***21***(6), pp. 829–837.
[4] Shivakumar, R., 2014. "How to tell which decisions are strategic". *California Management Review,* ***56***(3), pp. 78–97.
[5] Mayer, I. S., 2009. "The gaming of policy and the politics of gaming: A review". *Simulation and Gaming,* ***40***(6), pp. 825–862.
[6] Grogan, P. T., and Meijer, S. A., 2017. "Gaming methods in engineering systems research". *Systems Engineering,* ***20***(6), pp. 542–552.

[7] Abt, C. C., 1987. *Serious Games*. University Press of America, Lanham, Maryland.

[8] March, S. T., and Smith, G. F., 1995. "Design and natural science research on information technology". *Decision Support Systems,* ***15***(4), pp. 251–266.

[9] Hazelrigg, G. A., 1998. "A framework for decision-based engineering design". *Journal of Mechanical Design,* ***120***(4), pp. 653–658.

[10] Suh, N. P., 1998. "Axiomatic design theory for systems". *Research in Engineering Design,* ***10***(4), pp. 189–209.

[11] Collopy, P. D., and Hollingsworth, P. M., 2011. "Value-driven design". *Journal of Aircraft,* ***48***(3), pp. 749–759.

[12] Zeigler, B. P., Praehofer, H., and Kim, T. G., 2000. *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*, second ed. Academic Press, San Diego, California.

[13] Grogan, P. T., and de Weck, O. L., 2015. "The ISoS modeling framework for infrastructure systems simulation". *IEEE Systems Journal,* ***9***(4), pp. 1139–1150.

[14] Ross, A. M., and Rhodes, D. H., 2008. "Using natural value-centric time scales for conceptualizing system timelines through epoch-era analysis". In INCOSE International Symposium, INCOSE, pp. 1186–1201.

[15] Ross, A. M., Rhodes, D. H., and Hastings, D. E., 2008. "Defining changeability: Reconciling flexibility, adaptability, scalability, modifiability, and robustness for maintaining system lifecycle value". *Systems Engineering,* ***11***(2), pp. 246–262.

[16] Sterman, J. D., 1994. "Learning in and about complex systems". *System Dynamics Review,* ***10***(2–3), pp. 291–330.

[17] Kalyuga, S., Ayres, P., Chandler, P., and Sweller, J., 2010. "The expertise reversal effect". *Educational Psychologist,* ***38***(1), pp. 23–31.

[18] Davis, P. K., 1988. Applying artificial intelligence techniques to strategy-level gaming and simulation. Tech. Rep. N-2752-RC, The RAND Corporation, Santa Monica, California.

[19] Davis, P. K., and Henninger, A., 2007. Analysis, analysis practices, and implications for modeling and simulation. Tech. Rep. OP-176-OSD, The RAND Corporation, Santa Monica, California.

[20] Endsley, M. R., 1995. "Toward a theory of situation awareness in dynamic systems". *Human Factors,* ***37***(1), pp. 32–64.

[21] Schell, J., 2014. *The Art of Game Design: A book of lenses*. CRC Press, Boca Raton, FL.

[22] Harteveld, C., 2011. *Triadic game design: Balancing reality, meaning and play*. Springer, London, UK.

[23] Grogan, P. T., and de Weck, O. L., 2015. "Interactive simulation games to assess federated satellite system concepts". In IEEE Aerospace Conference, IEEE.

[24] Grogan, P. T., Ho, K., Golkar, A., and de Weck, O. L., 2016. "Multi-actor value modeling for federated systems". *IEEE Systems Journal,* ***PP***(99), pp. 1–10.

[25] Ren, Y., Bayrak, A. E., and Papalambros, P. Y., 2016. "EcoRacer: Game-based optimal electric vehicle design and driver control using human players". *Journal of Mechanical Design,* ***138***(6), p. 061407.

[26] Sexton, T., and Ren, M. Y., 2017. "Learning an optimization algorithm through human design iterations". *Journal of Mechanical Design,* ***139***(10), p. 101404.

[27] Kim, N., Cha, S., and Peng, H., 2011. "Optimal control of hybrid electric vehicles based on Pontryagin's minimum principle". *IEEE Transactions on Control Systems Technology,* ***19***(5), pp. 1279–1287.

[28] Wei, X., Guzzella, L., Utkin, V., and Rizzoni, G., 2007. "Model-based fuel optimal control of hybrid electric vehicle using variable structure control systems". *Journal of Dynamic Systems, Measurement, and Control,* ***129***(1), pp. 13–19.

[29] Sutton, R. S., and Barto, A. G., 1998. *Reinforcement learning: An introduction*, Vol. 1. MIT Press, Cambridge, Massachusetts.

[30] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al., 2015. "Human-level control through deep reinforcement learning". *Nature,* ***518***(7540), p. 529.

[31] Kang, N., Feinberg, F. M., and Papalambros, P. Y., 2017. "Autonomous electric vehicle sharing system design". *Journal of Mechanical Design,* ***139***(1), p. 011402.

[32] Zadbood, A., and Hoffenson, S., 2017. "Agent-based modeling of automobile producer and consumer behavior to support design for market systems analysis". In ASME 2017 International Design Engineering Technical Conferences, ASME. Paper No.V02AT03A041.

[33] Shiau, C.-S. N., Michalek, J. J., and Hendrickson, C. T., 2009. "A structural analysis of vehicle design responses to corporate average fuel economy policy". *Transportation Research Part A: Policy and Practice,* ***43***(9-10), pp. 814–828.