Information Exchange Patterns in Digital Engineering: An Observational Study Using Web-Based Virtual Design Studio

Joseph Thekinen[1]
Collective Design Lab School of Systems and Enterprises,
Stevens Institute of Technology,
Hoboken, NJ 07030
e-mail: thekinen@msu.edu

Paul T. Grogan
Collective Design Lab School of Systems and Enterprises,
Stevens Institute of Technology,
Hoboken, NJ 07030
e-mail: pgrogan@stevens.edu

This paper performs an observational human subjects study to investigate how design teams use an information system to exchange, store, and synthesize information in an engineering design task. Framed through the lens of decision-based design, a surrogate design task models an aircraft design problem with 12 design parameters across four roles and six system-level functional requirements. A virtual design studio provides a browser-based interface for four participants in a 30-min design session. Data collected across 10 design sessions provide process factors about communication patterns and outcome factors about the resulting design. Correlation analysis shows a positive relationship between design iteration and outcome performance but a negative relationship between chat messages and outcome performance. Discussion explains how advances in information exchange, storage, and synthesis can support future design activities.
[DOI: 10.1115/1.4050087]

Keywords: computational foundations for engineering optimization, engineering informatics, human computer interfaces/interactions, model-based systems engineering, decentralized multidisciplinary engineering design process modeling

1 Introduction

The recent push for digitalization and adoption of information systems (ISs) in engineering design seeks to improve the efficiency and effectiveness of design organizations [1,2]. Intelligent digital design systems help carry out parallel and synergistic activities to enhance design quality and success rate of one-time development [3]. Pursuit of greater automation generates increased interest in design science perspectives in IS research, specifically on system design methods, human–computer interfaces, information retrieval, and information exchange algorithms [4,5].

Digitalization transforms design activities in several ways: replacing paper-based requirements documents with information systems, streamlining design activities between geographically distributed design teams with web-based platforms [6–8], replacing higher fidelity tools with physics-based meta-models [9], and integrating organizationally dispersed tools with novel computational architectures [9]. This transformation affects designer activities and, in turn, their behavior, particularly in multidisciplinary design where designers have limited knowledge outside of their discipline. Designing a complex artifact involves the exchange of information between design actors with diverse expertise and geographically separated by large distances. Design decisions in one subsystem affect the constraints and parameters of others, obfuscated by limited knowledge. Advances in disciplinary analysis further aggravate the problems caused by this knowledge gap [8].

To take full advantage of digitalization, IS capabilities must support design actor behavior and design organization characteristics. However, generalizable knowledge about how IS are used in a design context is understudied because of the difficulty of conducting studies in operational environments. To address this gap, we study how digitalization affects designer communication strategies and resulting process efficiency in a prototypical virtual design studio. Results of an observational study across 10 design sessions suggest that adapting two concepts—value-driven design and agile design—to industrial design operations may yield benefits.

This paper frames an aircraft design problem as a multi-level and multidisciplinary design activity using a web-based IS. An observational behavioral experiment studies the effect of IS on resulting design activities and outcomes. This study seeks knowledge and principles to improve the efficiency of systems engineering and design activities with advanced IS. We extend results and observations to evaluate new features to improve information exchange, develop new tools such as computational design assistants, and implement new design methods to improve design team performance.

2 Background

2.1 Engineering Design. Tayal [10] defines engineering design as an iterative decision-making process in which basic science, mathematics, and engineering sciences convert resources to meet stated objectives. Iterations span multiple disciplines in large-scale design problems that require a wide range of expertise [11].

Disciplines are defined in terms of knowledge areas or components. For example, aircraft design disciplines possess knowledge in functional areas such as aerodynamics, propulsion, and control or components such as wing, tail, and fuselage. There could be hundreds of such disciplines depending on the organizational architecture [12]. Individual disciplines are handled by subsystems and systems designers who coordinate the activities of the subsystems.

Subsystems are often interdependent and coupled by the physics of the design problem. Design decisions in one subsystem depend on the design parameters controlled by other subsystems [13]. Subsystems also have limited knowledge of how design decisions affect constraints and parameters of other subsystems.

Information exchange between subsystems can help overcome interdependencies and knowledge gaps. However, design systems face communication and organization challenges as barriers to smooth information exchange. Communication challenges include

[1]Present address: School of Planning Design and Construction, Michigan State University, East Lansing, MI 48823.

Manuscript received September 17, 2020; final manuscript received February 2, 2021; published online February 23, 2021. Assoc. Editor: Matthew I. Campbell.

heterogeneous computing environments and high communication costs [8]. Organization challenges include geographic barriers between subsystems and institutional constraints on free information flow [12]. Hindered information flow results in schedule overruns, increased design cost, and project delays which adversely affect the competitive edge [8].

Studying interdependencies and interactions between system elements from technical and behavioral perspectives can help address challenges to information exchange. Technical studies view system elements as design tasks or product components to focus on characteristics such as chronological task order, design complexity, and coupling between disciplines. Behavioral studies view system elements as design actors to focus on human factors such as balancing cognitive load, minimizing bias, and preventing strategic behavior.

One class of technical studies models coupling between design tasks driven by physical or logical connectivity or resource exchange in designed components which can be reduced by decreasing feedback and information exchange requirements. The design structure matrix (DSM) is a popular matrix representation of interdependencies between design tasks [14]. Smith and Eppinger [15] developed a Work Transformation Model to predict which features in an automotive brake system will require many iterations. DSM algorithms such as clustering, partitioning, sequencing, and tearing aim to reorder the matrix elements according to some criteria such as reducing feedback, information exchanges, and coupling [14,16,17]. Reducing feedback characteristics of interdependent tasks in concurrent engineering results in less iterations [18]. Similarly, reducing coupling between design tasks decreases the number of design cycles [19]. Simply changing the content of some design tasks—such as decomposing coupled tasks or changing task specifications—decreases the number of potential iterations by as much as 50% in aerospace applications [20]. Information technology, such as systems that predict implications of design modifications on other subsystems, could hypothetically help decompose coupled tasks but broader implications on the behavior of design actors are not well understood.

Another class of technical studies streamlines communication of system objectives and balances subsystem optimization to improve efficiency. Value-driven design (VDD) structures system engineering to communicate system objective functions down to each component and maintain balances between tradeoffs [21]. Cheung et al. [22] show how VDD enables rational decisions to be made at every level of engineering design using aero-engines as context. Castagne et al. [23] show VDD concepts that allow the manufacturer to develop more efficient aircraft fuselage designs with higher profit and operator gains. However, VDD requires a human value judgment to synthesize a scalar value measure from system attributes and behavior of subsystem design actors in response to such information is not well understood.

Multidisciplinary design optimization (MDO) studies numerical optimization techniques to design engineering systems with multiple disciplines [12,24,25]. Martins and Lambe [12] compare MDO architectures using a common framework to visualize data dependency and information flow between computational components. Although MDO literature extensively studies distributed architectures to solve coupled design problems, the sole focus is to improve iteration efficiency assuming a given behavior of the design actors. The manifestation of distributed architectures on the iteration and information exchange behavior of design actors is not well-understood. For example, some disciplines in aircraft design are more computationally expensive than others [26], leading to asymmetric delays and failures to complete design task in the allotted time. Communication pattern and designer behavior in such a heterogeneous distributed environment are not understood.

Behavioral studies focus on design actor characteristics, cognitive loads from design task allocations, and effects of information availability. Chuadari et al. [27] investigated how the design cost and task complexity affect the process-level information acquisition decisions made by human designers. Hirschi and Frey [28] studied the effect of cognitive load on problem-solving in coupled parameter design problems, showing evidence that limited working memory significantly limits designer efficiency. These behavioral studies focus on the effect of characteristics of design problem such as coupling and complexity on designer behavior. The effect of information technology on designer behavior is not well understood, even though it has been shown that information hiding may lead to inefficiencies in iterative cycles described as design churn [29].

2.2 Information Systems for Engineering Design. Information systems are key knowledge management components of engineering design that facilitate interaction and information exchange between members of a team [30]. Domain experts from several disciplines need to exchange information efficiently to prevent divergence of local engineering models [31]. Engineering tools and data are specific to a discipline, and it is challenging to share heterogeneous engineering data with other disciplines. Factors at actor-, project-, and company-level organizations influence shared understanding in collaborative design [32]. Parallel to general advances in IS technology, applications of IS to engineering design cross several paradigms.

Concurrent engineering (CE) is an integrated design methodology matured in the 1990s to shorten lead times, improve customer satisfaction, and reduce costs by minimizing communication delays between disciplinary designers through co-location and IS infrastructure [33]. The five pillars of modern CE include the team (people), a model (shared design knowledge), tools (software applications), the process (sequence of activities), and the facility (supporting infrastructure, including IS) [34]. The aerospace domain exhibits the strongest adoption of CE during conceptual design due to the strong coupling between disciplines [35]. IS infrastructure for CE exchanges design parameters between disciplines, commonly via macro-enabled spreadsheets, and aggregates and tracks resulting system attributes over time.

Efforts to improve model interoperability in the late 1990s, including automation for MDO activities, developed IS technology to integrate disciplinary physics-based models. For example, the MODELICA modeling language facilitates exchange of models and model libraries with high adoption in the automotive domain [36]. The associated Functional Mock-up Interface (FMI) standard allows dynamic model interaction using C-language code, linked binaries, and an XML configuration file [37]. Other co-simulation architectures compose discrete or continuous time models [38]. Distributed co-simulation standards such as IEEE 1278 Distributed Interactive Simulation (DIS), and IEEE 1516 High Level Architecture (HLA) leverage computer networks to exchange information via standard protocols.

More recent interest in model-based systems engineering (MBSE) applies IS platforms to a broader scope of design activities by establishing a single-integrated model shared across a design organization [39]. Modeling languages such as the integrated computer-aided manufacturing (ICAM) definition for functional modeling (IDEF0), and object-process methodology (OPM), and systems modeling language (SysML) provide graphical representations of a shared model. Supporting IS infrastructure includes commercial and open source platforms to retrieve, view, edit, and save shared models. For example, the Open Model-based Engineering Environment (MBEE) provides model development kits, model management databases, and view editors to facilitate information exchange [40].

Despite progress over the past 30 years, widespread adoption of IS platforms for integrated modeling across engineering design as a field remains sparse. MBSE itself is in the early stages and carries many holdover activities (and culture) from traditional systems engineering that may not match future design processes [41]. In contrast to the centrally defined IS architectures such as MODELICA and SYSML that require specific modeling languages or software, service-oriented architectures (SOAs) describe an enterprise data management strategy that coordinates information exchange using "loosely coupled, reusable, standards-based services" with data interoperability rather than application interoperability [42]. Similar to other IS applications such as web services, SOA may help accommodate increasing scale and complexity necessary in digital engineering applications, even by simply "wrapping" or abstracting existing co-simulation techniques [43].

2.3 Research Gap. While many new IS platforms have been proposed ranging from CE to MBSE, there is a general lack of understanding of how information technology influences communication behavior of the design actors, particularly in a multi-level and multi-disciplinary design environment. On one hand, IS reduces the cost of communication to alleviate barriers; however, it also provides pervasive connectivity that may strengthen interdependencies. A stronger theoretical basis for the relationship between IS capabilities and engineering design outcomes could help inform future investments in IS technology with fewer costly trials and experiments.

To address this gap, our primary research question (RQ) asks: *how do IS-enabled communication strategies such as information exchange frequency, storage, and synthesis relate to process efficiency for decentralized design?* The process efficiency is measured as improvement rate in design quality, where a higher quality corresponds to a design closer to meeting system requirements.

Our research approach collects and analyzes data from an observational human subject study conducted on a web-based digital design platform. The design platform hosts a scaled-down surrogate aircraft design task that aligns with industry design operations to support transfer of insights to inform technology investment. We structure the design task to collect observational data while capturing technical information exchange between decentralized design roles. This activity also created a distributed software architecture with a multi-user interface that controls access to design parameters and analysis models to mimic design activities of geographically distributed multi-disciplinary design teams.

Data collection during the observational study records design team activities, interactions, and outcomes. Correlation analysis identifies relationships between process and outcome variables of interest. Discussion aligns observations with design principles such as VDD, agile design, and version control in MBSE.

3 Aircraft Design As a Surrogate Design Task

To advance research objectives, this section explains the foundations, formulation, and implementation of an electric aircraft design problem used as a surrogate design task.

3.1 Abstract Model of Engineering Design. Building on a foundation of decision-based design [44] and axiomatic design theory [45], a design task is composed of N input design parameters (DPs) $x = \{x_1, \ldots, x_N\}$ and M functional requirements (FRs) $y = \{y_1, \ldots, y_M\}$ related to system performance attributes and evaluated by a system model $y = F_y(x)$. Traditional requirements-based design activities pursue a set of R binary requirements $z = \{z_1, \ldots, z_R\}$, $z_i \in \{0, 1\}$ related to FRs through a requirements function $z = F_z(y)$. For example, the requirement

$$z_i = \begin{cases} 1 & y_j \geq y_j^\star \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

verifies if scalar FR y_j meets required threshold $y_j^\star$.

The static relationship between DPs and FRs can be characterized by an $N \times M$ design matrix $\mathbf{D}$ where element $d_{ij} \approx \partial y_j / \partial x_i$ measures the sensitivity of FR y_j to DP x_i. Binary design matrices only characterize the presence (1) or absence (0) of interactions. An $N \times N$ design structure matrix (DSM) $\mathbf{M}$ characterizes interdependencies between DPs where element $m_{ij} \approx \partial x_j / \partial x_i$, subject to iso-performance outcomes (i.e., identical FRs). Multiplying the design matrix with its transpose $\mathbf{M} = \mathbf{D} \times \mathbf{D}^T$ provides an undirected physical domain DSM derived from functional dependencies [46].

Systems engineering assigns control and visibility of DPs and FRs (respectively) to members of a design organization of S actors. The binary $S \times N$ control matrix $\mathbf{C}$ with elements $c_{ij} \in \{0, 1\}$ indicates if actor i controls DP x_j. Typically, each DP is only controlled by one actor such that $\sum_i c_{ij} = 1$. A binary $S \times M$ visibility matrix $\mathbf{V}$ with elements $v_{ij} \in \{0, 1\}$ indicates if actor i views FR y_j. A $S \times S$ social dependency matrix $\mathbf{S} = \mathbf{C} \times \mathbf{D} \times \mathbf{V}^T$ characterizes interdependencies among design actors where element s_{ij} indicates actor j depends on information from actor i.

The design process follows an iterative sequence with an initial design vector $x^{(0)}$ and subsequent vectors $x^{(t)}$ after design period $t > 0$. First, design actors work exclusively within assigned DPs and FRs. For example, actor k modifies controlled DPs $x_k = \{x_i : c_{ik} = 1\}$ and observes effects on visible FRs $y_k = \{y_i : v_{ik} = 1\}$ using a local subsystem model $y_k = F_{y,k}(x_k, x_{-k})$ where $x_{-k} = \{x_i : c_{ik} = 0\}$ represents dependent DPs controlled by other design actors (superscripts omitted for clarity). Note that x_{-k} requires explicit information exchange with other actors to receive updated values. After design iteration t, system-level integration composes DPs from each design actor to yield $x^{(t)} = \{x_k\}$ with associated FRs $y^{(t)} = F_y(x^{(t)})$. The iterative sequence repeats with subsystem-level modifications and system-level synthesis.

3.2 Aircraft Design Task. Aircraft design is an example of large-scale coupled design problems. It exhibits a large design space and tightly-coupled decisions across subsystem boundaries, demanding systems engineering to coordinate technical analysis. For example, the thrust requirements of the propulsion subsystem are dependent on the aerodynamic drag produced by fuselage and airfoils. Data from each subsystem must be readily available to inform design of dependent subsystems.

Aircraft design typically consists of three phases: conceptual design, preliminary design, and detailed design, differing in the detail of design definition. This task targets the preliminary design phase which aims to choose design parameter values for a selected concept (i.e., parameterization). It is an iterative process consisting of an inner loop and an outer loop for each iteration [8]. Figure 1 shows a schematic of the design activities during the inner loop and outer loop for each design iteration. The inner loop performs disciplinary analysis (structural, aerodynamic, propulsion, controls, etc.) to achieve subsystem-level design requirements. Once converged, an outer loop performs system-level analysis (experimental testing or computer simulations) on the complete aircraft. If the system-level requirements are not achieved, the whole process is repeated in a new iteration.

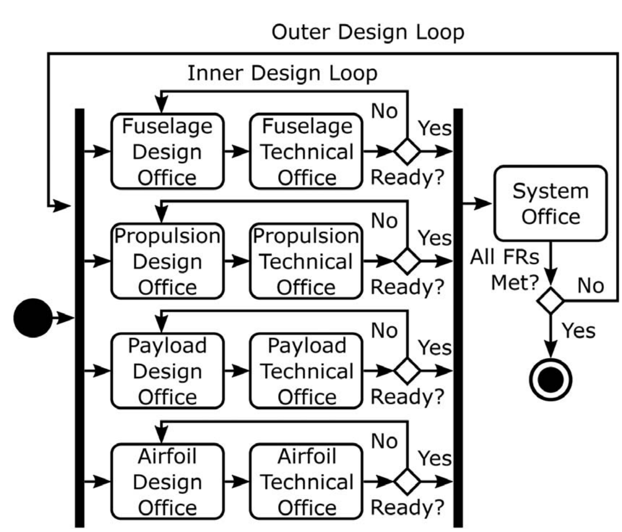

Fig. 1 Activity diagram with inner and outer design iteration loops for aircraft preliminary design

Following the abstract task definition in Sec. 3.1, this study develops an electric aircraft design task abstracted into a modular form Ref. [47] with four subsystems: fuselage, payload (battery), propulsion (motor and propeller), and airfoil (wing and tail). The task defines 12 DPs, 12 subsystem-level FRs, and six system-level FRs tied to design requirements. Table 1 presents a binary design matrix **D** showing dependencies between DPs (rows) and FRs (columns). Although modular at the subsystem level, the task is highly integral at the system level due to numerous interdependencies (typical of aerospace domain problems).

The task is allocated to a design organization with four actors and five roles: four *subsystem roles* specific to each actor and one *systems role* shared among the four. Subsystem roles operate design offices that control assigned DPs. Each design office has a corresponding technical office to perform technical analysis to yield visible subsystem FRs. Technical offices use computational models such as Vortex Lattice Method and Blade Element Theory, empirical relations, and semi-analytic equations described in Appendix B.

The system role operates a system office that collects information from all subsystems to analyze six additional FRs. Six corresponding system-level requirements:

$$z_1 = \begin{cases} 1 & y_{13} \geq 30\,\text{min} \\ 0 & \text{otherwise} \end{cases} \qquad z_2 = \begin{cases} 1 & 0 \leq y_{14} \leq 10\,\text{N} \\ 0 & \text{otherwise} \end{cases}$$

$$z_3 = \begin{cases} 1 & 0 \leq y_{15} \leq 10\,\text{N} \\ 0 & \text{otherwise} \end{cases} \qquad z_4 = \begin{cases} 1 & -5 \leq y_{16} \leq 5\,\text{N} - \text{m} \\ 0 & \text{otherwise} \end{cases}$$

$$z_5 = \begin{cases} 1 & y_{17} \geq 0\,\text{m}^3 \\ 0 & \text{otherwise} \end{cases} \qquad z_6 = \begin{cases} 1 & y_{18} \leq 30\text{USD} \\ 0 & \text{otherwise} \end{cases}$$

seek to meet a minimum endurance, constrain flight dynamical (lift, thrust, moment) and physical (volume) properties to valid regions, and meet a cost threshold.

3.3 Virtual Design Studio. The VIRTUAL DESIGN STUDIO, developed as part of this research, is the distributed software application that hosts design experiments for the aircraft design task as a prototype of IS infrastructure. The overall design philosophy is to provide a simple and extensible multi-user interface accessible to users. The key capabilities required to support experiments include the following:

(1) Concurrent use by geographically distributed design team members
(2) Technical information exchange between designer roles
(3) Simple and intuitive user interface controls and feedback
(4) 3D renderings or visualizations of design concepts
(5) Interface with automated execution of technical analysis models in PYTHON
(6) Automatic logging of all user actions for experimental observation and post-processing.

The software architecture in Fig. 2 follows a layered service-oriented architecture with three main components: (1) front-end/client services, (2) middleware/application services, and (3) back-end/technical services. The layers are connected using HyperText Transfer Protocol (HTTP) requests, containerized using Docker virtualization, and deployed/hosted on cloud computing instances.

The front-end services provide a browser-based client to designers and administrators during experimental design sessions. Designer communication with the application is secured with a SSL connection (i.e., HTTPS) via a reverse proxy. Each actor controls a design interface (Fig. 3(*a*)) which provides control over the constituent DPs, domain-specific outputs of technical analyses (including 3D visualizations for airfoil and fuselage components), buttons to trigger the analysis outputs (Update) and advance to the next round of design iteration (Ready), and a chat-based messaging system. Each actor also has access to a system office (Fig. 3(*b*)), which shows all the system-level FRs and visualization of the whole aircraft.

During a design session, design actors independently modify and update their assigned DPs and supporting FRs during a design period of up to 150 s. After this period (or when all four designer roles signal "Ready"), the interface switches to the system-level outputs for review of system FRs, constraint limits, and satisfied requirements. Finally, when all designers signal "Ready" to start the next round, the interface switches back to the domain-specific components for the next round of design iteration.

Middleware services connect the front-end with the back-end to provide user authentication, maintenance of the application state, and orchestration of technical services to support a design task. Service endpoints update the design task, toggle the application mode and timer state, broadcast chat messages, accept new DP values, and disseminate technical analysis results as FR values. To improve application responsiveness, a caching system stores commonly accessed results (e.g., system-level technical analyses required for all designer clients at the end of a design round) in memory.

Back-end technical services contain all of the engineering models needed for technical analyses. It leverages OPENVSP[2] and SUAVE [48], two domain-specific modeling software tools for aircraft design and analysis. OPENVSP offers a parametric environment to generate and modify aircraft 3D geometry and produce visualizations as X3D files. SUAVE provides aerodynamic-, structural-, and mission-specific analysis of design configurations generated using OPENVSP. Both provide a PYTHON API so the back-end services wrap the underlying functionality as a web service. Notably, the OPENVSP PYTHON API is not thread-safe so simultaneous requests to dependent technical analyses can cause concurrent modification. To mitigate this technical limitation, the middleware uses asynchronous locks to ensure only one active back-end service request for OPENVSP at a time.

3.4 Design Task Assumptions and Limitations. While closely following the abstract model of engineering design, the proposed task has a few limitations. First, it is representative of large-scale systems engineering tasks only at an abstract level. It specifies only four design actors, rather than the hundreds normally involved in preliminary design studies (thus, each actor models more than one individual). The task unfolds over a period of approximately 30 min, rather than weeks or months, and assumes rapid technical analysis feedback and instant chat-based communication. No domain experience is assumed, and the technical analyses contain many simplifications and approximations. Additionally, no systems engineering requirements are provided at the subsystem level such that design actors must implement their own strategy to achieve system-level requirements.

Despite the purposeful simplifications, we tailored the aircraft design task to need several design iterations to meet stated design requirements, yet still be solvable in an experimental timescale of about 1 h including training. It was purposefully structured to have strong subsystem coupling, similar to integral systems in aerospace applications. We carefully chose task dimensions such as the problem size (number of DPs) and number of subsystems to tailor task complexity to cognitive and communication abilities in the virtual design studio. Finally, designer actions to run technical office and system office analyses and communicate via the text-based chat system demand significant resources relative to the limited time availability, permitting study of how communication strategies affect outcomes of IS-enabled design teams. While the mode of communication does not influence outcomes in some studies [49], other factors such as synchronicity may have stronger impacts on distributed teams [50] and should be investigated in further studies.

[2]http://www.openvsp.org

Table 1 Binary design matrix for the aircraft design task

		Fuselage			Payload			Propulsion			Airfoil			Aircraft (system)					
Subsystem	Design parameter	Mass (y_1)	Drag (y_2)	Volume (y_3)	Energy (y_4)	Volume (y_5)	Mass (y_6)	Power (y_7)	Mass (y_8)	Thrust (y_9)	Lift (y_{10})	Drag (y_{11})	Mass (y_{12})	Endurance (y_{13})	Net Lift (y_{14})	Net Thrust (y_{15})	Net Moment (y_{16})	Net Volume (y_{17})	Cost (Mass) (y_{18})
Fuselage	Length (x_1)	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
Fuselage	Max diameter (x_2)	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
Fuselage	Wing location (x_3)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Payload	Number cells (x_4)	0	0	0	1	1	1	0	0	0	0	0	0	1	1	0	1	1	1
Payload	Charge per Cell (x_5)	0	0	0	1	1	1	0	0	0	0	0	0	1	1	0	1	1	1
Payload	Location (x_6)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Propulsion	Diameter (x_7)	0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	0	0	1
Propulsion	Rotation rate (x_8)	0	0	0	0	0	0	1	0	1	0	0	0	1	0	1	0	0	0
Propulsion	Number blades (x_9)	0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	0	0	1
Airfoil	Wing span (x_{10})	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0	1
Airfoil	Root chord length (x_{11})	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0	1
Airfoil	Tail scaling (x_{12})	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0	1

4 Observational Study on Communication Strategies

The virtual design studio that hosts surrogate aircraft design task provides a platform to conduct human subject studies. This section reports the procedure, collected data, and observations from an initial exploratory human subject study of observed communication strategies.

4.1 Observational Study Overview. This study aims to understand the correlative relationship between communication strategies observed across design groups and associated design outcomes. There are no control variables in the study, which limits the ability to determine causality between input/process factors and outcome factors. However, we observe factors associated with high- or low-performing teams for further study.

We conducted 10 design sessions, each with four participants (one per design office). The participants work in a co-located setting but communication is only permitted through the text-based chat interface during the design task (i.e., no speaking). Participants perform subsystem- and system-level design iterations until either they achieve the design goal of meeting all six system-level FRs (as described in Sec. 3.2) or run out of time. In the subsystem-level loop, design actors update DPs of their assigned design office and view updated subsystem FRs from the technical office which takes less than 1 s to compute and retrieve. In the system-level loop, the design actors receive the system office analysis results synthesizing information from all the subsystems. During this time, the design actors review system-requirements, plan the next steps, and delegate tasks for the next iteration.

4.2 Data and Construct Operationalization. Automated logs collect the time-stamp, input DPs, and output FRs for each design office update (design action), time-stamp and content of all chat messages, and duration of each subsystem- and system-level iteration for each session.

Chat messages, design actions, design iterations, and the system-level time fraction are process variables that measure various dimensions of the observed communication strategies. Chat messages measure direct communication between design offices while design iterations and time in the system-level loop measure indirect communication via technical analysis feedback that provides information about the state of other subsystems. The design iteration rate measures the frequency that analysis results are available to participants. The time spent on the system-level loop measures the temporal resources allocated to interpret system-level analysis results.

A system-level normalized error metric measures design quality after each iteration based on the distance (in FR space) from meeting all requirements normalized by the variation in each FR. Appendix A provides details including the equation and normalization constants such that normalized error ranges between 0 and 1. A design that meets all requirements has normalized error of zero.

4.3 Study Protocol. We conducted ten sessions by recruiting a pool of 40 student participants. The minimum eligibility for recruitment included at least junior standing in a science/technology/engineering/mathematics degree program and a prior engineering design experience. A majority of recruited participants were graduate students with an undergraduate engineering degree. Incentives for participation include a $10 online retail gift card plus a bonus of $5, the design meets all system-level FRs before the end of the session.

Each session starts from a baseline design that meets two of the six requirements—endurance (z_1) and volume check (z_5). Appendix B shows the design parameter values of the baseline design. A 1-h session proceeds as follows:

- 10 min: arrival and informed consent.
- 2 min: pre-recorded video (for consistency across sessions) that describes the virtual design studio, their roles, and experiment roles.
- 8 min: self-exploration of the web interface.

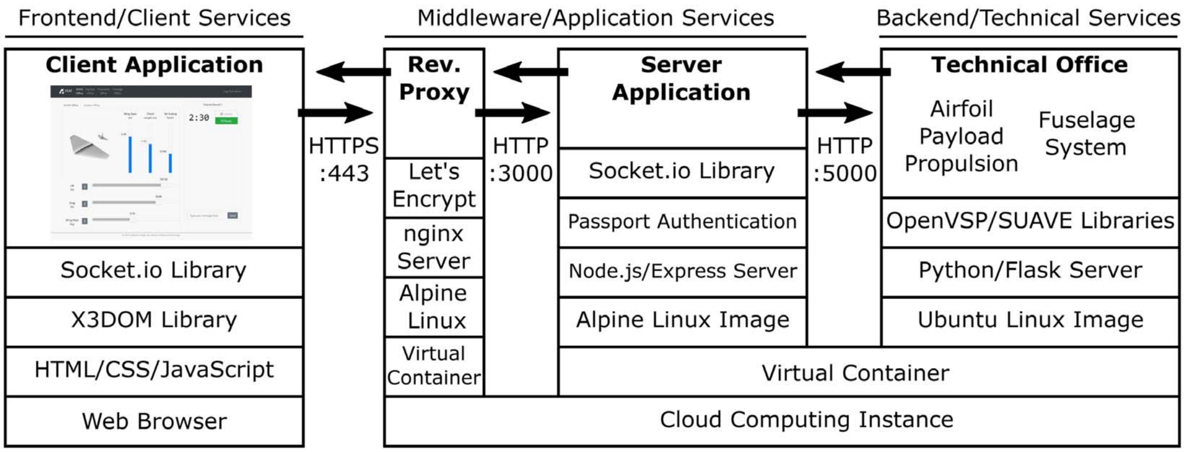

Fig. 2 Virtual design studio software architecture including front-end, middleware, and back-end services

- 10 min: tutorial design task that mimics design processes such as timed design iterations and chats.
- 30 min: experimental aircraft design task.

Participants maintain the same design role for tutorial and experimental tasks to build familiarity with assigned DPs and FRs. The tutorial and experimental design tasks have different cruising speeds to allow general learning about the problem domain without transferring specific solution DPs.

We do not set constraints on the maximum number of messages or design iterations. However, a maximum time of 150 s is enforced in the subsystem-level and system-level of each iteration such that each complete iteration lasts at most 300 s (5 min) and at least six design iterations occur in 30 min. All four participants must mutually agree to end a design iteration before the maximum allotted 150 s, signaled by each participant clicking the "Ready" button on their interface.

5 Results and Analysis

This section reports observed results, starting with an overview of process and outcome variables in Sec. 5.1. Next, Sec. 5.2 analyzes the correlation between process and outcomes variables observed across the 10 sessions. Finally, Sec. 5.3 further investigates the temporal dynamics of process and outcome variables.

5.1 Session Overview.

Table 2 summarizes the process metrics related to communication strategies and outcome metrics for design quality in each session. Only two sessions (4 and 6) achieved the design goal of meeting all six system-level requirements. Session 6 completed the fastest and was labeled the highest-performing session. Session 8 was labeled the lowest-performing as they produced the design with the lowest quality and only met two system-level requirements. Inter-team variation is attributed to factors such as differences in attitude/behavior, communication, leadership, and trust [32] but not necessarily differences in shared understanding, disrupted routines [51], or knowledge representation [31] because of the surrogate task's simplified and synthetic nature.

Figure 4 shows the 3D visualization of the baseline and final designs for each session to illustrate the breadth of geometries pursued. Figure 4(*a*) shows the baseline aircraft. Figures 4(*e*) and 4(*g*) show the final aircraft geometries from the sessions that produced designs that met all system-level FRs. Following the foundation in domain-specific tools SUAVE and OPENVSP, geometries that resemble traditional aircraft generally provide higher design

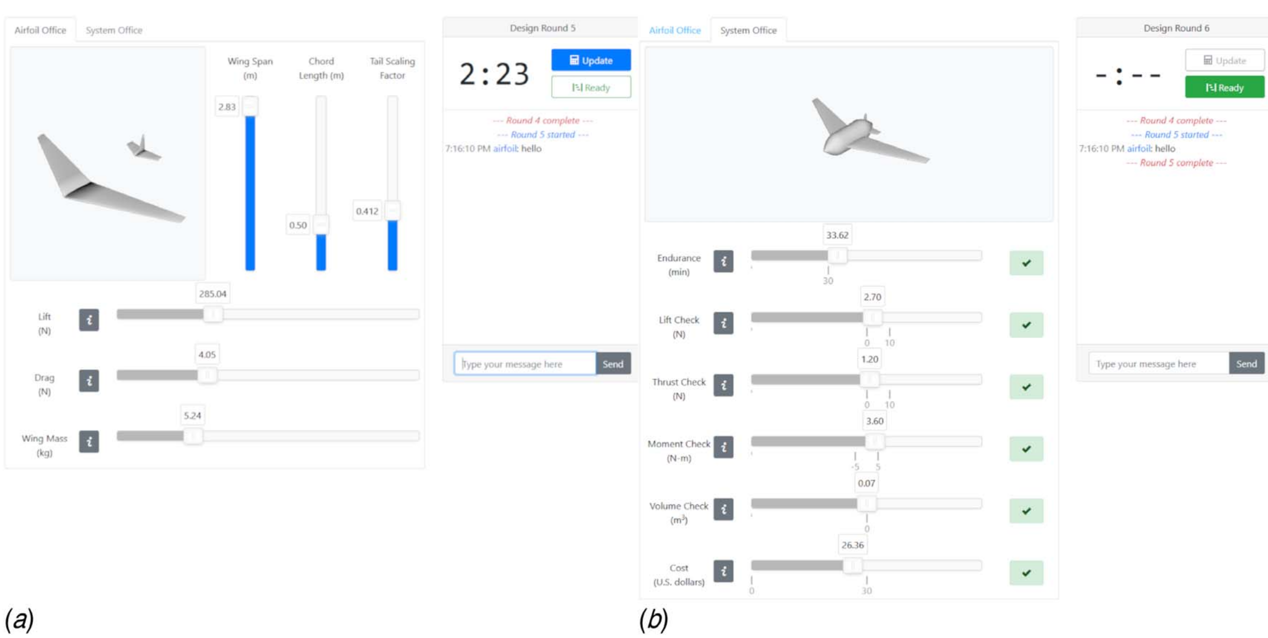

Fig. 3 Front-end client for design actors with assigned DPs (horizontal sliders) and FRs (horizontal sliders): (*a*) airfoil design office and (*b*) system office

quality, even though geometry only captures a subset of DPs. Figures 4(*b*) and 4(*i*) show the final aircraft geometries produced by sessions with the highest normalized error.

While a detailed analysis of chat messages is beyond the scope of this study, we selected messages from one successful (Session 6) and one failed session (Session 8) to help give some context and flavor to the following analyses. Tables 5 and 6 in Appendix C show text messages in the first and last few design iterations for Sessions 6 and 8. A few qualitative observations can be made observing text messages from Sessions 6 and 8. In general, we observe that the more successful session had: (a) a higher quality of the text, with fewer grammatical and spelling errors, (b) more system-focused and goal-directed communication, and (c) a higher pressure to hit ready and complete more iterations during the allotted session time. For both sessions, shorter and more directed text messages appeared toward the end of the session.

5.2 Correlation Analysis. Figure 5 shows a panel of four scatter plots that visualize observed relationships between process and outcome variables. Nonparametric correlation analysis using Spearman's rank correlation coefficient (r_s, computed with SciPy 1.4.1 function `stats.spearmanr`) quantifies the magnitude and significance of acausal relationships. An associated p-value based on a two-tailed hypothesis test estimates statistical significance with $\nu = 8$ degrees -of-freedom. While the small number of samples limits the strength of conclusions, results indicate some trends for additional investigation.

Figure 5(*a*) compares the rate of design iterations (number iterations divided by session duration) with the associated session performance via normalized error. Correlation analysis shows a negative Spearman's rank correlation coefficient of $r_s = -0.626$ with a moderate level of significance ($p = 0.053$). Therefore, evidence suggests quicker design iterations are correlated with lower normalized error and better performance. Limited to acausal inference, this result cannot rule out other factors that contribute both to higher iteration rates and improved performance.

Figure 5(*b*) compares the fraction of time in the system office with the resulting normalized error. Recall the default timer allots 150 s for both the design office and system office phases, yielding a reference system office time fraction of 0.5 which can be altered as participants advance between phases. Half of the sessions spent more than 50% of the overall time in the system office phase. Correlation analysis shows a negative Spearman's rank correlation coefficient of $r_s = -0.438$ where the larger system office time fractions are correlated with lower normalized error (and improved performance); however, the limited statistical significance ($p = 0.206$) cannot support strong claims.

Figure 5(*c*) compares the rate of chat messages exchanged (total messages divided by session duration) with the resulting normalized error. Correlation analysis shows a positive Spearman's rank correlation coefficient of $r_s = 0.632$ with moderate level of significance ($p = 0.050$). Evidence suggests a larger volume of messages exchanged is correlated with lower design quality and session performance, contrary to expectations. Possible explanations could stem from the primitive nature of the chat interface that limits effective communication or reliance on the chat interface for teams that struggle to conceptualize the task.

Finally, Fig. 5(*d*) compares the rate of design actions performed (total actions divided by session duration) with the resulting session performance via normalized error. Correlation analysis shows a nearly zero Spearman's rank correlation coefficient with low significance ($r_s = 0.036$, $p = 0.920$) that indicates no relationship exists between the frequency of DP changes and design outcomes.

5.3 Process and Outcome Variable Time Series. The preceding analysis only investigates process and outcome metrics at the conclusion of a session. To get a better understanding of the internal dynamics during a session, Figure 6 shows a panel of four time series plots to illustrate the evolution of process and outcome variables.

Figure 6(*a*) shows most sessions maintain a constant design iteration rate with a slight increase at the end, attributed to session deadline boundary effects. Several sessions maintain iteration rates near the minimum (5 min per iteration). Figure 6(*b*) shows most sessions exchange messages at a constant rate, and Session 3 exchanges significantly fewer messages compared to all other sessions. Figure 6(*c*) shows a general decreasing design action rate during a session and distinct patterns between design office phases (when actions accumulate) versus system office phases (when no actions are possible).

The normalized error per design iteration in Fig. 6(*d*) shows a general downward trend as teams progress during the design task; however, improvement is not monotonous for most sessions and some sessions do not submit the best design. Figure 7 compares the best observed design at any iteration to the final submitted design showing that six of the ten sessions did not end with the highest quality design.

6 Discussion and Recommendations

Based on results from the exploratory study, this section discusses potential opportunities to improve design performance by augmenting the IS platform. The discussion is organized around information exchange and storage/synthesis as two essential functions supported by IS. Additional reflection comments on limitations of the surrogate design task and broader implications for design in industry settings.

6.1 Information Exchange. The virtual design studio exhibits three major types of information exchange: (1) design office feedback about how subsystem DPs impact subsystem FRs, (2) system office feedback about how subsystem DPs impact system-

Table 2 Summary of observed process and outcome metrics from ten study sessions

	Process			Outcome			
Session	System office %	Iterations	Messages	Actions	Req. Met	Norm. Err.	Duration[a]
1	63.2	13	123	202	2	0.065	29:58.2
2	48.8	7	172	199	3	0.024	30:03.7
3	35.0	9	30	220	4	0.006	29:45.0
4	55.6	14	121	217	6	0.0	30:09.1
5	43.2	15	154	236	3	0.039	30:04.4
6	59.3	17	105	142	6	0.0	26:55.1
7	66.4	21	177	85	5	0.001	30:00.1
8	35.6	8	183	181	2	0.084	30:03.0
9	53.4	8	155	109	5	0.039	30:02.2
10	41.9	7	182	212	3	0.080	29:32.0

[a]Max session duration differs slightly from 30:00 due to experimental conditions.

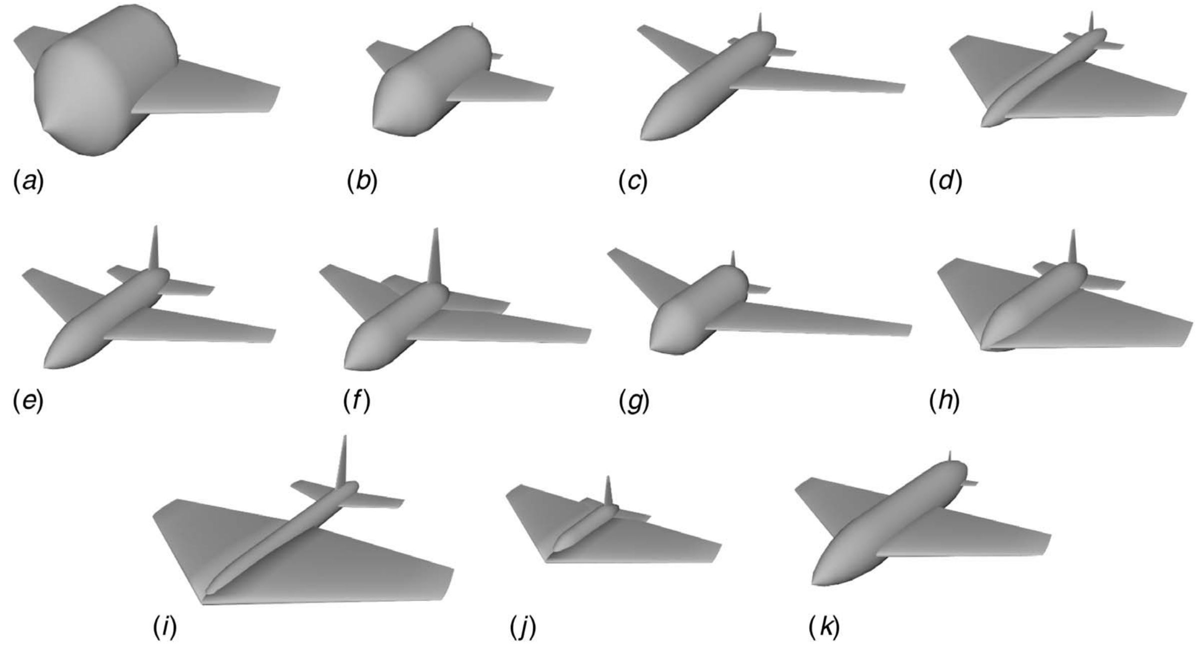

Fig. 4 Comparison of baseline and final design geometries for each session (session 4 and 6: design meets all requirements): (*a*) baseline, (*b*) session 1, (*c*) session 2, (*d*) session 3, (*e*) session 4 , (*f*) session 5, (*g*) session 6, (*h*) session 7, (*i*) session 8, (*j*) session 9, and (*k*) session 10

level FRs, and (3) chat messages exchanged between subsystems. Observations of interactions and communication patterns guide how and where to focus resources to improve effectiveness of a digital design platform. Augmented communication protocols seek to improve coordination and efficiency in decentralized design activities.

6.1.1 Feedback and Communication Channels. Design teams have freedom to choose how to allocate their time and attention to various communication channels. Throughout a design session, a team alternates between the design office phase to set DPs and view subsystem-level FRs and the system office phase to view system-level FRs. Figure 5(*b*) shows a large variation in system

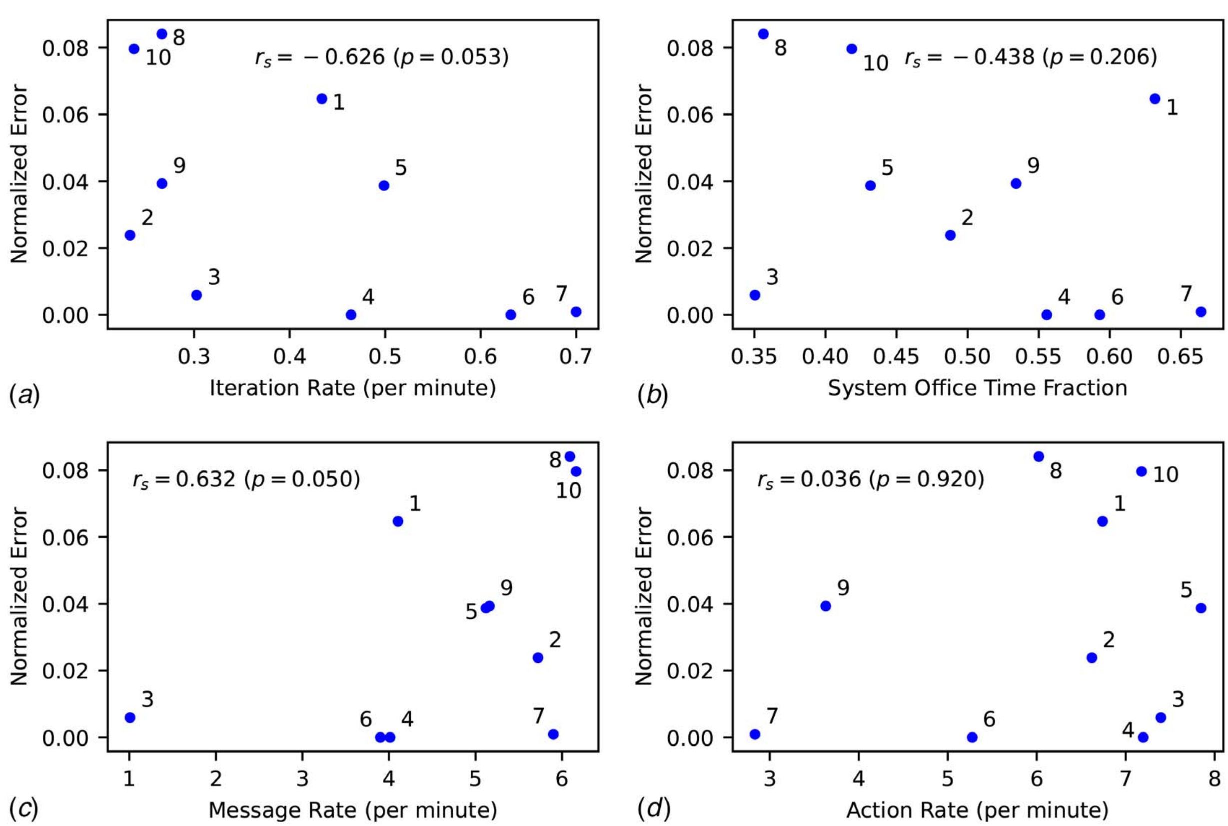

Fig. 5 Correlation analysis between process and outcome variables observed (numbered by session)

office allocations ranging from 35.0% to 66.4% and a weak positive correlation with session performance. The system office phase relays important feedback but also restricts designer modifications. Future extensions could investigate improved channels to communicate system-level information or otherwise focus designer efforts during the design office phase.

Chat messages are a less formal but immediate (i.e., outside the design cycle) mode of information exchange, similar to e-mails or telephone calls in practice. In general, information exchange in any channel consumes limited time resources. Here, participants face a tradeoff between allocating time to compose chat messages or interacting with the design office. Figure 5(*c*) shows a negative correlation between the message rate and session performance, suggesting chat does not uniformly support design activities. However, the lack of causality poses a major barrier to interpretation because it is plausible that other individual- or team-level factors contribute both to increased message rates (e.g., in attempt to establish a common understanding of the problem) and lower overall performance. Additional evidence is required to understand if reducing the cost of this communication channel by, for example, allowing voice communication, contributes to improved session performance.

6.1.2 Design Iteration Frequency. Design iteration provides repeated feedback during a session. Correlation analysis in Fig. 5(*a*) indicates higher iteration rates are associated with improved session performance. Additionally, Fig. 6(*a*) provides evidence that the default timer may anchor the selected iteration rate near 0.2 iterations per minute for some sessions. Augmenting the interface with a lower default timer value may encourage quicker design cycles and potential improvement in session performance.

Eliciting faster design cycles aligns with agile design principles which refer to making rapid changes in response to expected or unexpected environmental changes [52]. Design agent failure to meet requirements is one of the three distinct scenarios that require an agile reaction to ensure timely production [52]. This scenario aligns with observations that shorter design cycles facilitating small continuous changes with more frequent design reviews and feedback are associated with better design outcomes.

Frequent design iteration in the surrogate design task helps to identify and resolve major dependencies across subsystem design offices to achieve system-level FRs. Observations indicate certain FRs, such as the moment balance, are sensitive to subsystem DPs and require iteration to resolve concurrent modification. This factor is likely amplified in the surrogate task as participants lack deep disciplinary experience and formal change modification processes; however, interdependencies also contribute to design churn in industry applications [29]. In absence of a more structured change management methodology, frequent iteration helps to isolate the antecedent conditions of system-level FRs.

In industrial applications, faster design cycles translate to a shorter time interval between critical design review phases but also must recognize that early cycles reflect an incomplete design concept. Practical logistics, organizational culture, overhead on iteration cycles, and cognitive timescales limit the maximum viable design cycle rate.

6.2 Information Storage and Synthesis. The virtual design studio provides essential storage and synthesis capabilities to record the current state of each DP and compute and display the value of each FR. Even with a timer, Fig. 7 shows several sessions failed to submit the "best" design in terms of normalized error. This observation could be attributed to an inability to easily revert to a previous design state or from cognitive difficulty in evaluating and comparing holistic quality across iterations. The following sections discuss the potential benefit of storing a historical record of DP changes and synthesizing a system-level metric to summarize the high-dimensional FR space.

6.2.1 State History. Extending the scope of the virtual design studio to store the history of design states, in addition to the current state, allows individual design offices to revert to a previous state. With the addition of a state history, a designer has four choices during design process: revert DPs, update DPs, send a chat message, or signal the end of an iteration. Reverting to a previous design state may disrupt process continuity by accessing outdated dependent information but judicious use could enable broader exploration without losing a promising intermediate design state. Storing multiple design variants enables tradespace exploration following design-of-experiments or a set-based principles that could further automate some search activities, subject to computational limitations.

IS-enabled design state history aligns with version control in MBSE applications, despite current challenges for automatic and universal adherence [53]. Providing a single repository with version control is often impractical for inter-disciplinary organizations. Alternatively, providing different repositories and tools with version control tailored to characteristics of each subsystem e.g., PLM system for CAD/CAM, SCM system for software code as a "Multiple Local Repository Approach" allows team members within an organization use their individual computers to store versions they develop [53]. Outstanding challenges to be addressed include timely notification of version changes and synchronization of subsystem models.

6.2.2 System Metric. It is common for complex systems to have hundreds or thousands of requirements. Tracking progress is often limited to a few critical metrics such as system mass, balance, or cost. The surrogate aircraft design task has comparatively few system-level FRs (six) but still presents a challenge to quantify how "good" one set of DPs is from another. Synthesizing a simple system-level quality measure as a scalar number would help designers from all subsystem disciplines evaluate the collective system-level performance. For example, the normalized error metric used in analysis could be operationalized in the design task to provide a scalar quality indicator. While mapping multiple FRs to a scalar number loses some information, it would ease the cognitive load. We do not quantify or measure the cognitive load on the subjects in our study. In our approach, we tweaked the cognitive load so that the design task was doable in 30 min.

The concept of using a scalar metric as the basis for design decisions aligns with elements of VDD. In VDD, subsystems receive system-level information through value functions, equivalent to the normalized error metric discussed here. In MBSE, requirements engineering supports activities such as requirements elicitation from end users, requirements allocation, requirements management, and requirements validation as key components of such a quality metric [54]. The use of normalized metric as a formal representation will help convey end-user requirements to subsystem designers and monitor its progress.

6.3 Revisiting the Research Question. We found information exchange frequency to have mixed effects on session performance depending on the type of information exchange. While a higher frequency of chat messages correlated with lower overall performance, frequent design iteration, and system office feedback enhanced session performance, the conclusions are from student participants under scaled-down experimental conditions. There needs to be an additional study on whether these results translate to industrial settings with expert designers. External factors such as experience level of designers and time-scale of the experiment may influence observations: a steeper learning curve may amplify the positive influence of higher design iterations on novice designers and a limited session duration may amplify the negative effect of chat time. The inferences and observations are from an experiment with a text-based chat interface. Audio- or video-based chat interface might change frequency, volume, and quality of information

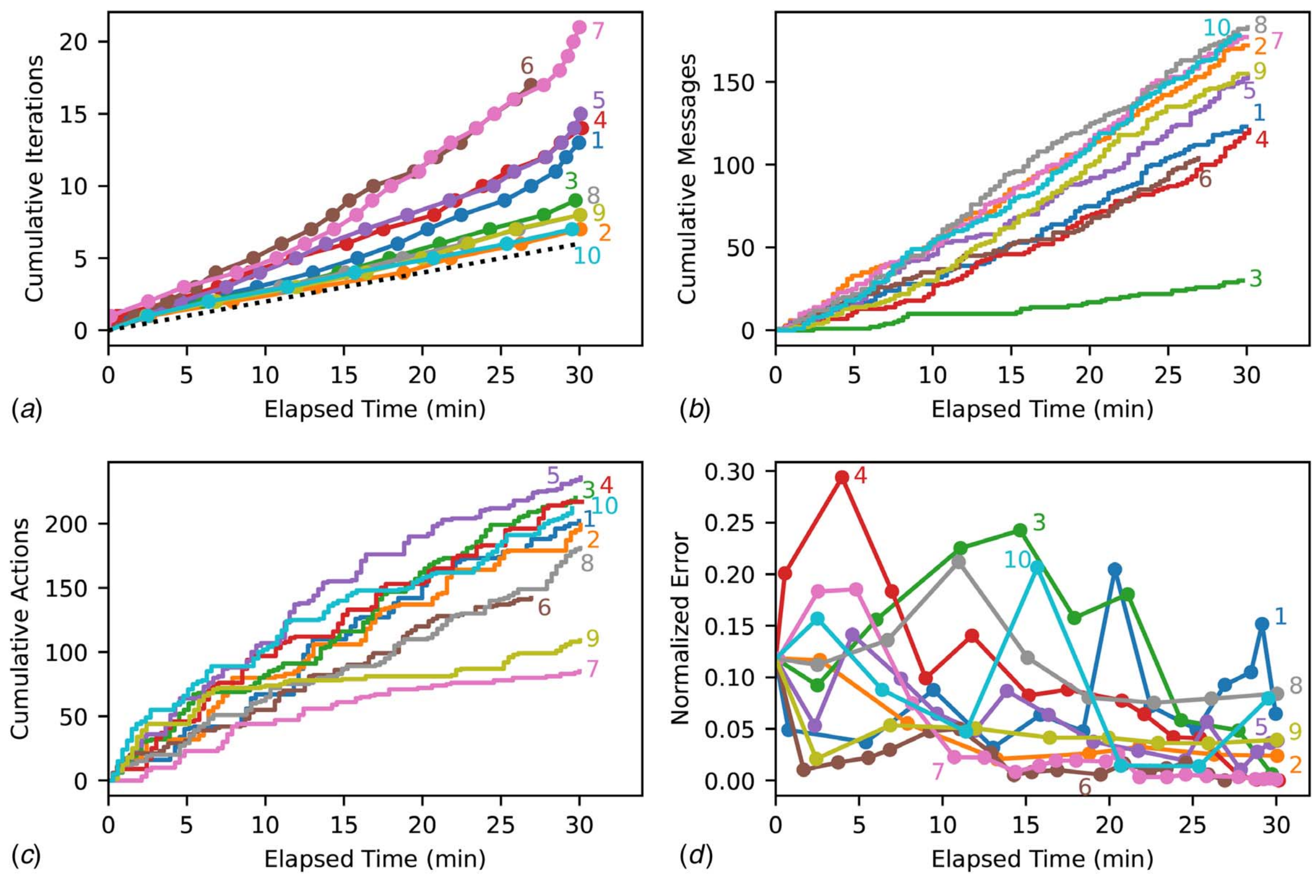

Fig. 6 Process and outcome variable time series during each session (numbered by session)

exchanged. The effects of audio- or video-based chat interface are not understood and is a limitation of this study.

The recommendation that information storage and synthesis improve session performance that is from the observation that a majority of sessions failed to submit their best design. This recommendation ignores the fact that the sessions had a hard deadline of 30 min, after which the participants were not allowed to make any design updates. Alternative augmentations such as softening the deadline by providing additional time, in the end, may improve performance. Besides, the effects of cognitive factors and design continuity on information storage and synthesis need further study.

Despite the limitations of the study, the virtual design studio and the surrogate design task provide a basic approach for future studies on the effects of digital information systems on design performance, particularly in decentralized and web-based applications. The observations and conclusions from a scaled-down lab study with student participants resembled design principles such as agile design, value-driven design, and version control in MBSE.

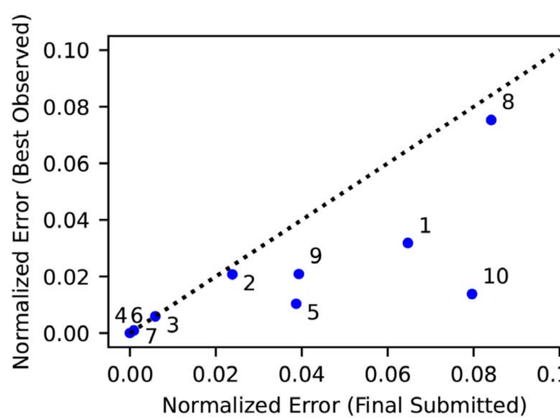

Fig. 7 Comparison of the best design observed in each session to the final submitted design (numbered by session). The dashed line indicates if the best design were the final.

7 Conclusion

Engineers rely on IS-enabled design platforms to alleviate cognitive and communication boundaries in engineering design. However, knowledge of how IS technology influences design communication and behavior remains limited due to the difficulty of collecting data in operational design settings. This paper formulates a surrogate aircraft design problem that can be completed by a team of four non-expert disciplinary participants in approximately 30 min and implements a prototype IS platform as a browser-based virtual design studio. While simplified, the surrogate task builds on a foundation of decision-based design and shares structural similarities to complex design problems.

The observational study collected process- and outcome-oriented data from ten sessions of student design teams. While causality cannot be determined for some significant relationships, results provide evidence of important connections between design iteration rate, message rate, and outcome performance. Observing individual sessions shows several sessions fail to submit their best designs despite having an on-screen timer to track the submission deadline.

The results in this paper provide an early and imperfect insight into how information systems influence communication and behaviors among design teams. While there is a desire for future work to adopt stronger control over input conditions, this will likely remain difficult to the high variability in individual and team capability and high cost of collecting observational data.

Observations from student participants show evidence for potential IS features to improve team performance: reducing the cost of communication channels, anchoring a more frequent design iteration rate (similar to agile design processes), providing a state history of past design configurations (similar to version control in MBSE), and synthesizing a scalar system metric to quantify design quality (similar to VDD). While some of these features resemble design principles, future work needs to explore their effectiveness. While tailored to the surrogate design task in this study, the virtual design studio may help support future research by providing a browser-based environment to conduct distributed design sessions.

Nomenclature

t = design iteration number
s = session number
x = set of design parameters
y = set of functional requirements
z = set of binary requirements
$\mathbf{C}$ = control matrix (c_{ij} is 1 if actor i controls x_j)
$\mathbf{D}$ = design matrix (d_{ij} is sensitivity of y_j to x_i)
$\mathbf{M}$ = design structure matrix (m_{ij} is sensitivity of x_j to x_i)
$\mathbf{S}$ = social dependency matrix (s_{ij} is 1 if j depends on i)
$\mathbf{V}$ = visibility matrix (v_{ij} is 1 if actor i views y_j)
M = number of functional requirements
N = number of input design parameters
S = number of design actors
R = number of binary requirements
x_i = ith design parameter
y_i = ith functional requirement
z_i = ith binary requirement
$x^{(t)}$ = set of design parameters after iteration t
E^s = normalized system error in session s
E^s_t = normalized system error after tth iteration in session s
R^s_t = system requirements met after tth iteration in session s
$F_y(x)$ = model to evaluate functional requirements
$F_z(y)$ = model to evaluate binary requirements

Acknowledgment

This research was completed with funding support from the Lockheed Martin Corporation (Reference S18-006).

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The authors attest that all data for this study are included in the paper. Data provided by a third party listed in Acknowledgment.

Appendix A: Normalized Error Calculation

Equation (A1) computes the normalized error for iteration t of session s for current FR values $y^s_{i,t}$ with feasible range given by lower bound $y_i^{(L)}$ and upper bound $y_i^{(U)}$, and normalization factor $y_i^{(N)}$.

$$E^s_t = \frac{1}{6}\sum_{i=13}^{18} \frac{\max\left(y^s_{i,t}, y_i^{(U)}\right) - y_i^{(U)} + \min\left(y^s_{i,t}, y_i^{(L)}\right) - y_i^{(L)}}{y_i^{(N)}} \tag{A1}$$

Table 3 Normalized error constants

Functional Req.	$y_i^{(N)}$	$y_i^{(L)}$	$y_i^{(U)}$
Endurance (y_{13})	570.7 min	30 min	9999 min
Net lift (y_{14})	55.62 N	0 N	10 N
Net thrust (y_{15})	464.58 N	0 N	10 N
Net moment (y_{16})	618.31 N m	−5 N m	5 N m
Net volume (y_{17})	6.4 m^3	0 m^3	9999 m^3
Cost (mass) (y_{18})	38.93 USD	0 USD	30 USD

The normalization factor corrects for scaling differences between FRs (e.g., net moment, y_{16}, has an order of magnitude higher variability than net lift, y_{14}). Table 3 provides $y_i^{(N)}$, $y_i^{(L)}$, and $y_i^{(U)}$ for each FR.

Appendix B: Technical Office Details

This appendix describes the equations and analysis models for subsystem- and system-level FRs for the aircraft design task. Figure 8 illustrates a wireframe model of the baseline aircraft modeled using OPENVSP with corresponding DP values in Table 4.

Fuselage Design Office. The fuselage design office controls the geometry of the tapered cylindrical vessel that houses the payload via the length (x_1, m), max diameter (x_2, m), and wing location (x_3, % fuselage). Subsystem FRs include mass (y_1, kg), drag (y_2, N), and volume (y_3, m^3).

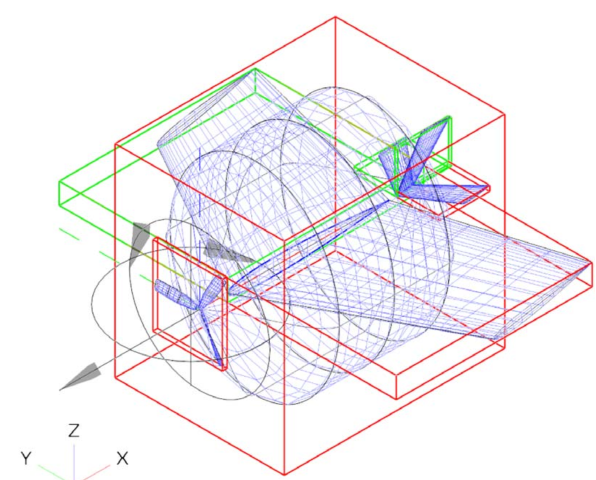

Fig. 8 Baseline aircraft wireframe model

Table 4 Baseline aircraft design parameters

Subsystem	Design parameter	Value
Fuselage	Length (x_1)	1.57 m
Fuselage	Max diameter (x_2)	0.86 m
Fuselage	Wing location (x_3)	50 %
Payload	Number cells (x_4)	8
Payload	Energy per cell (x_5)	4.1 A h
Payload	Location (x_6)	58.72 %
Propulsion	Diameter (x_7)	0.63 m
Propulsion	Rotation rate (x_8)	2900 rpm
Propulsion	Number blades (x_9)	2
Airfoil	Wing span (x_{10})	2.39 m
Airfoil	Root chord length (x_{11})	1.15 m
Airfoil	Tail scaling (x_{12})	2.0

Fuselage drag is calculated with empirical models. At constant cruising speed, the zero-lift drag coefficient is given by [47]

$$C^f_{D_o} = K\left(1 + \frac{60}{(x_1/x_2)^3} + \frac{0.0025}{x_1/x_2}\right)\frac{x_1}{x_2}$$

where K is a constant independent of x_1 and x_2, estimated from CFD analysis of the fuselage geometry. From $C^f_{D_o}$, drag is given by

$$y_2 = \frac{1}{2}\rho_{air} v^2 S^f_{ref} C^f_{D_o}$$

where $\rho_{air} = 1.225\,\text{kg/m}^3$ is the density of the air at the cruising altitude, v is the aircraft cruising speed, and S_{ref} is the reference area of the geometry which we estimate using panel methods in SUAVE. The fuselage mass and volume

$$y_1 = \pi x_1 x_2 \rho_{skin} \rho_{thk}$$

$$y_3 = \pi x_1 \left(\frac{x_2}{2}\right)^2$$

approximate the fuselage as a hollow cylinder with thickness $\rho_{thk} = 3.175\,\text{mm}$ and material density $\rho_{skin} = 1700\,\text{kg/m}^3$.

Payload Design Office. The payload design office manages the battery system that provides power to the propulsion system via designing the number of cells (x_4), charge per cell (x_5, A h), and location (x_6, % fuselage). Subsystem FRs include available energy (y_4, W h), required volume (y_5, m^3), and mass (y_6, kg).

Available energy is given by

$$y_4 = x_4 x_5 V_{cell}$$

where $V_{cell} = 3.7\,\text{V}$ is the battery cell voltage. Payload mass is given by

$$y_6 = x_4 m_{cell}$$

where $m_{cell} = 0.057\,\text{kg}$ is the battery cell mass. Payload volume y_5 is given by

$$y_5 = y_6/\rho_{cell}$$

where $\rho_{cell} = 3.8\,\text{kg/m}^3$ is the battery cell density.

Propulsion Design Office. The propulsion design office manages the propeller selection and corresponding power demands based on the propeller diameter (x_7, m), rotation rate (x_8, rpm), and number of blades (x_9). Subsystem FRs include required power (y_7, W), mass (y_8, kg), and generated thrust (y_9, N).

We import propeller geometry into SUAVE and use CFD (blade element theory) to calculate the thrust and torque (denoted by M^p) generated by the propeller at cruising speed [55]. The CFD model assumes a viscosity of $1.5 \cdot 10^{-5}\,\text{N-s/m}^2$. The required power is given by

$$y_7 = M^p x_8$$

as a function of the rotation rate. We assume propeller mass increases linearly with the number of blades and the square of propeller diameter.

$$y_8 = 0.25 x_7^2 x_9$$

Airfoil Design Office. The airfoil design office manages the geometry of the wing and tail surfaces based on the wing span (x_{10}, m), root chord length (x_{11}, m), and tail scaling factor (x_{12}). Subsystem FRs include lift (y_{10}, N), drag (y_{11}, N), and mass (y_{12}, kg).

OPENVSP wing geometry is imported to SUAVE [48] to calculate the lift $C^w_L(i)$ and drag coefficients $C^w_D(i)$ using CFD analysis (vortex lattice method) and estimate the platform or reference area $S^w_{ref}(i)$ using wing panel methods for each aerodynamic element i (i.e., wings and tail components). Lift and drag forces are calculated as

$$y_{10} = \sum_i \frac{1}{2}\rho_{air} v^2 S^w_{ref}(i) C^w_L(i)$$

$$y_{11} = \sum_i \frac{1}{2}\rho_{air} v^2 S^w_{ref}(i) C^w_D(i)$$

where $v = 22.22\,\text{m/s}$ is the cruising speed. Wing mass is estimated as

$$y_{12} = \sum_i \frac{1}{2}\rho_{air}\rho_{skin}\rho_{thk} S^w_{surf}(i)$$

where $S^w_{surf}(i)$ is the wing airfoil surface area estimated from the wing geometry in SUAVE, $\rho_{skin} = 1700\,\text{kg/m}^3$ is the mass density, and $\rho_{thk} = 3.175\,\text{mm}$ is the skin thickness.

System Office. The six system-level FRs include endurance (y_{13}, min), net lift (y_{14}, L), net thrust (y_{15}, N), net moment (y_{16}, N-m), net volume (y_{17}, m^3), and cost (y_{18}, USD).

Endurance (in minutes) is given by

$$y_{13} = \frac{y_4}{y_7}$$

assuming that propulsion consumes all of the energy provided by the payload. Net lift is given by

$$y_{14} = y_{10} - 9.81(y_1 + y_6 + y_8 + y_{12})$$

as the difference between lift generated by all airfoil surfaces and the weight force, ignoring fuselage lift as it is typically less than 5% of the total lift for the geometry considered in our model [47].

Net thrust is given by

$$y_{15} = y_9 - (y_2 + y_{11})$$

as the difference between propeller thrust and total drag, ignoring drag induced by the interaction between the wing and fuselage. The net moment, computed about the aircraft nose for simplicity, is given by

$$y_{16} = M^w_{pitch} - 9.81 x_1\left(y_6\frac{x_6}{100} + y_{12}\frac{x_3}{100} + y_1\frac{1}{2}\right) + x_1\frac{x_3}{100} y_{10}(1 + 0.2 x_{12})$$

where M^w_{pitch} is the pitching moment generated by the wing and tails obtained from CFD analysis. While calculating moment, we assume that a tail with tail-scaling factor (x_{12}) of 1 produces 20% of the moment generated by the wings. The net volume is given by

$$y_{17} = y_3 - y_5$$

as the difference between available and required volume. Finally, cost is given by

$$y_{18} = \rho_{cost}(y_1 + y_6 + y_8 + y_{12})$$

assuming $\rho_{cost} = 1.0$ USD/kg.

Appendix C: Chat Messages

Table 5 Chat message exchanges during first and last few iterations in session 6

Time	Iter.	Sender: message
00:03	0	PP: payload please move the wings
00:34	1	PP: how much power do we have in hand?
00:45	1	PP: give me a number
00:58	1	PL: 970.9
01:22	1	PL: volume is 3.12 and mass is 11.87
01:24	1	PP: I need around 1500
01:32	1	AF: Hit ready for max updations
01:42	1	PL: i cant increase it
02:00	1	PP: a little more power and we are good
02:14	1	PL: our cost is almost right
02:45	1	FL: How can we affect the endurance?
02:55	1	PP: power increase would solve it
24:27	15	PL: done
24:44	15	PP: moment is increasing again
24:49	15	FL: I will increase volume a little bit more
24:50	15	PL: what's wrong with moment
24:58	15	AF: idk
25:00	15	PP: payload is too low
25:05	15	PP: wing is big
25:30	15	PP: hit ready
25:49	16	PL: guys hit ready fast
25:53	16	PL: ?
26:02	16	PP: wow
26:08	16	PL: volume
26:30	16	PL: whoever changed moment pls increase it slightly
26:53	17	PL: guys hurry
26:58	17	AF: wowowowoowow

Note: AF, Airfoil; PP, propulsion; PL, payload; FL, fuselage.

Table 6 Chat message exchanges during first and last few iterations in session 8

Time	Iter.	Sender: message
00:05	1	PP: ok so i think we need to have a basis, once you do it, click on update, and then let everyone on this group know the values
01:00	1	PP: fuselage drag now is 22.66
01:14	1	PP: Power is 3283
01:14	1	PL: what about wing drage
01:24	1	PL: 6.6
01:31	1	PP: who has energy
01:45	1	AF: energy charge 5.3
02:04	1	PL: payload 6.82
02:11	1	PP: 5.3?
02:14	1	PL: I decrease it a little
28:14	6	FL: now f drag 39.04
28:43	6	PL: payload location 75 and max
28:45	6	AF: wing drag 9
28:55	6	FL: thrust?
29:05	6	PP: 48 now
29:08	6	FL: match
29:09	6	FL: good
29:23	6	FL: lift?
29:33	6	AF: 610
29:57	6	AF: go for it, if you smaller it

Note: AF, airfoil; PP, propulsion; PL, payload; FL, fuselage.

References

[1] Glaessgen, E., and Stargel, D., 2012, "The Digital Twin Paradigm for Future NASA and US Air Force Vehicles," 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Paper No. AIAA-2012-1818.

[2] Reitenbach, S., Vieweg, M., Becker, R., Hollmann, C., Wolters, F., Schmeink, J., Otten, T., and Siggel, M., 2020, "Collaborative Aircraft Engine Preliminary Design Using a Virtual Engine Platform, Part A: Architecture and Methodology," AIAA Scitech 2020 Forum, Paper No. AIAA-2020-0867.

[3] Zhou, J., 2013, "Digitalization and Intelligentization of Manufacturing Industry," Adv. Manufact., **1**(1), pp. 1–7.

[4] Kuechler, W., and Vaishnavi, V., 2008, "The Emergence of Design Research in Information Systems in North America," J. Design Res., **7**(1), pp. 1–16.

[5] Hevner, A., and Chatterjee, S., 2010, Design Science Research in Information Systems, R. Sharda, ed., Springer, Boston, MA, pp. 9–22.

[6] Wang, H., Johnson, A., Zhang, H., and Liang, S., 2010, "Towards a Collaborative Modeling and Simulation Platform on the Internet," Adv. Eng. Inform., **24**(2), pp. 208–218.

[7] Larsson, T., Larsson, A., and Karlsson, L., 2001, "A Modular Approach to Web Based Multibody Dynamic Simulation," International CIRP Design Seminar Design in the New Economy, Stockholm.

[8] Reed, J. A., Follen, G. J., and Afjeh, A. A., 2000, "Improving the Aircraft Design Process Using Web-based Modeling and Simulation," ACM Trans. Model. Computer Simulation (TOMACS), **10**(1), pp. 58–83.

[9] Mavris, D. N., and Pinon, O. J., 2012, *An Overview of Design Challenges and Methods in Aerospace Engineering*, O. Hammami, D. Krob, and J. L. Voirin, eds., Springer, Berlin/Heidelberg, pp. 1–25.

[10] Tayal, S., 2013, "Engineering Design Process," Int. J. Computer Sci. Communicat. Eng., **18**(2), pp. 1–5.

[11] Kroo, I., Altus, S., Braun, R., Gage, P., and Sobieski, I., 1994, "Multidisciplinary Optimization Methods for Aircraft Preliminary Design," 5th Symposium on Multidisciplinary Analysis and Optimization, Paper No. AIAA-94-4325-CP.

[12] Martins, J. R., and Lambe, A. B., 2013, "Multidisciplinary Design Optimization: A Survey of Architectures," AIAA J., **51**(9), pp. 2049–2075.

[13] Grogan, P. T., and de Weck, O. L., 2016, "Collaboration and Complexity: An Experiment on the Effect of Multi-Actor Coupled Design," Res. Eng. Design, **27**(3), pp. 221–235.

[14] Steward, D. V., 1981, "The Design Structure System: A Method for Managing the Design of Complex Systems," IEEE Trans. Eng. Manag., **EM-28**(3), pp. 71–74.

[15] Smith, R. P., and Eppinger, S. D., 1997, "Identifying Controlling Features of Engineering Design Iteration," Manag. Sci., **43**(3), pp. 276–293.

[16] Eppinger, S. D., and Browning, T. R., 2012, *Design Structure Matrix Methods and Applications*, MIT Press, Cambridge, MA.

[17] Karniel, A., and Reich, Y., 2011, Managing the Dynamics of New Product Development Processes, Springer, London, UK, pp. 37–49.

[18] Eppinger, S. D., 1991, "Model-based Approaches to Managing Concurrent Engineering," J. Eng. Design, **2**(4), pp. 283–290.

[19] Kusiak, A., and Wang, J., 1993, "Efficient Organizing of Design Activities," Int. J. Prod. Res., **31**(4), pp. 753–769.

[20] Eppinger, S. D., 2001, "Innovation At the Speed of Information," Harvard Business Rev., **79**(1), pp. 149–158.

[21] Collopy, P. D., and Hollingsworth, P. M., 2011, "Value-Driven Design," J. Aircraft, **48**(3), pp. 749–759.

[22] Cheung, J., Scanlan, J., Wong, J., Forrester, J., Eres, H., Collopy, P., Hollingsworth, P., Wiseall, S., and Briceno, S., 2012, "Application of Value-Driven Design to Commercial Aeroengine Systems," J. Aircraft, **49**(3), pp. 688–702.

[23] Castagne, S., Curran, R., and Collopy, P., 2009, "Implementation of Value-Driven Optimisation for the Design of Aircraft Fuselage Panels," Int. J. Prod. Econ., **117**(2), pp. 381–388.

[24] Yi, S.-I., Shin, J.-K., and Park, G., 2008, "Comparison of Mdo Methods With Mathematical Examples," Struct. Multidiscipl. Optim., **35**(5), pp. 391–402.

[25] Simpson, T. W., Mauery, T. M., Korte, J. J., and Mistree, F., 2001, "Kriging Models for Global Approximation in Simulation-Based Multidisciplinary Design Optimization," AIAA J., **39**(12), pp. 2233–2241.

[26] Chittick, I. R., and Martins, J. R., 2009, "An Asymmetric Suboptimization Approach to Aerostructural Optimization," Optim. Eng., **10**(1), p. 133.

[27] Chaudhari, A. M., and Panchal, J. H., 2019, *An Experimental Study of Human Decisions in Sequential Information Acquisition in Design: Impact of Cost and Task Complexity*, A. Chakrabarti, ed., Springer, Singapore, pp. 321–332.

[28] Hirschi, N., and Frey, D., 2002, "Cognition and Complexity: An Experiment on the Effect of Coupling in Parameter Design," Res. Eng. Design, **13**(3), pp. 123–131.

[29] Yassine, A., Joglekar, N., Braha, D., Eppinger, S., and Whitney, D., 2003, "Information Hiding in Product Development: the Design Churn Effect," Res. Eng. Design, **14**(3), pp. 145–161.

[30] McMahon, C., Lowe, A., and Culley, S., 2004, "Knowledge Management in Engineering Design: Personalization and Codification," J. Eng. Design, **15**(4), pp. 307–325.

[31] Biffl, S., Lüder, A., Rinker, F., and Waltersdorfer, L., 2019, *Efficient Engineering Data Exchange in Multi-disciplinary Systems Engineering*, P. Giorgini, and B. Weber, eds., Springer, Rome, pp. 17–31.

[32] Liu, Y., Van Nederveen, S., and Hertogh, M., 2017, "Understanding Effects of Bim on Collaborative Design and Construction: An Empirical Study in China," Int. J. Project Manag., **35**(4), pp. 686–698.

[33] Pennell, J. P., and Winner, R. I., 1989, "Concurrent Engineering: Practices and Prospects," IEEE Global Telecommunications Conference and Exhibition.

[34] Knoll, D., Fortin, C., and Golkar, A., 2018, "Review of Concurrent Engineering Design Practice in the Space Sector: State of the Art and Future Perspectives," IEEE International Systems Engineering Symposium.

[35] Iwata, C., Infield, S., Bracken, J. M., McGuire, M., McQuirk, C., Kisdi, A., Murphy, J., Cole, B., and Zafirian, P., 2015, “Model-based Systems Engineering in Concurrent Engineering Centers,” AIAA SPACE 2015 Conference and Exposition.
[36] Mattsson, S. E., Elmqvist, H., and Otter, M., 1998, “Physical System Modeling With Modelica,” Control. Eng. Pract., **6**(4), pp. 501–510.
[37] Andersson, C., 2016, “Methods and Tools for Co-simulation of Dynamic Systems With the Functional Mock-Up Interface,” PhD thesis, Lund University, Lund, Sweden.
[38] Gomes, C., Thule, C., Broman, D., Larsen, P. G., and Vangheluwe, H., 2018, “Co-simulation: A Survey,” ACM Comput. Surveys, **51**(3), pp. 49:1–33.
[39] Ramos, A. L., Ferreira, J. V., and Barceló, J., 2012, “Model-based Systems Engineering: An Emerging Approach for Modern Systems,” IEEE Trans. Syst., Man, Cyber., Part C (Appl. Rev.), **42**(1), pp. 101–111.
[40] Kruse, B., and Blackburn, M., 2019, “Collaborating with OpenMBEE As An Authoritavite Source of Truth Environment,” Procedia Computer Science, Vol. 153, pp. 277–284.
[41] Madni, A. M., and Sievers, M., 2018, “Model-Based Systems Engineering: Motivation, Current Status, and Research Opportunities,” Syst. Eng., **21**(3), pp. 172–190.
[42] Andary, J. F., and Sage, A. P., 2010, “The Role of Service Oriented Architectures in Systems Engineering,” Inform. Knowledge Syst. Manage., **9**(1), pp. 47–74.
[43] Lu, J., Wang, J., Chen, D., Wang, J., and Törngren, M., 2018, “A Service-Oriented Tool-Chain for Model-Based Systems Engineering of Aero-Engines,” IEEE Access, **6**, pp. 50443–50458.
[44] Hazelrigg, G. A., 1998, “A Framework for Decision-Based Engineering Design,” ASME J. Mech. Des., **120**(4), pp. 653–658.
[45] Suh, N. P., 1998, “Axiomatic Design Theory for Systems,” Res. Eng. Design, **10**(4), pp. 189–209.
[46] Browning, T. R., 2016, “Design Structure Matrix Extensions and Innovations: A Survey and New Opportunities,” IEEE Trans. Eng. Manage., **63**(1), pp. 27–52.
[47] Sadraey, M. H., 2012, *Aircraft Design: A Systems Engineering Approach*, John Wiley & Sons, Chichester, U.K.
[48] Lukaczyk, T. W., Wendorff, A. D., Colonno, M., Economon, T. D., Alonso, J. J., Orra, T. H., and Ilario, C., 2015, “SUAVE: An Open-Source Environment for Multi-Fidelity Conceptual Vehicle Design,” 16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Paper No. AIAA-2015-3087.
[49] Ostergaard, K. J., Wetmore III, W. R., Divekar, A., Vitali, H., and Summers, J. D., 2005, “An Experimental Methodology for Investigating Communication in Collaborative Design Review Meetings,” CoDesign, **1**(3), pp. 169–185.
[50] Serçe, F. C., Swigger, K., Alpaslan, F. N., Brazile, R., Dafoulas, G., and Lopez, V., 2011, “Online Collaboration: Collaborative Behavior Patterns and Factors Affecting Globally Distributed Team Performance,” Comput. Human Behav., **27**(1), pp. 490–503.
[51] Abdirad, H., Dossick, C. S., Johnson, B. R., and Migliaccio, G., 2020, “Disruptive Information Exchange Requirements in Construction Projects: Perception and Response Patterns,” Build. Res. Inform., **49**(2), pp. 161–178.
[52] Matthews, P., Lomas, C., Armoutis, N., and Maropoulos, P., 2006, “Foundations of An Agile Design Methodology,” Int. J. Agile Manufact., **9**(1), pp. 29–38.
[53] Fisher, A., Nolan, M., Friedenthal, S., Loeffler, M., Sampson, M., Bajaj, M., VanZandt, L., Hovey, K., Palmer, J., and Hart, L., 2014, “3.1.1 Model Lifecycle Management for MBSE,” INCOSE International Symposium, Vol. 24, Wiley Online Library, Las Vegas, NV, pp. 207–229.
[54] Ryan, M., Cook, S., and Scott, W. R., 2013, “Application of MBSE to Requirements Engineering-research Challenges,” Faculty of Engineering and Information Sciences - Papers: Part A, no. 6335.
[55] MacDonald, T., Clarke, M., Botero, E. M., Vegh, J. M., and Alonso, J. J., 2017, “SUAVE: An Open-source Environment Enabling Multi-Fiidelity Vehicle Optimization,” 18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Paper No. AIAA-2017-4437.