Human-Centered Design Using System Modeling Language

Michael E. Watson
Christina F. Rusnock
John M. Colombi
Michael E. Miller

Recommended Citation
Human-Centered Design Using System Modeling Language
Michael E. Watson, Christina F. Rusnock, John M. Colombi, and Michael E. Miller, Air Force Institute of Technology

The human user is important to consider during system design. However, common system design models, such as the system modeling language, typically represent human users and operators as external actors, rather than as internal to the system. This research presents a method for integrating human considerations into system models through human-centered design. A specific system is selected to serve as the case study for demonstrating the methodology. The sample system is analyzed to identify the task and information flow. Then, both system- and human-centered diagrams are separately created to represent different viewpoints of the system. These diagrams are compared and analyzed, and new diagrams are created that incorporate both system and human considerations into one concordant representation of the system model. These new views allow systems engineers and human factors engineers to effectively communicate the role of the user during early system design trades.

Keywords: human systems integration, design methods, system dynamic analysis, robotics, military, command and control

INTRODUCTION

Human-centered design (HCD) holds the promise of creating system designs that are more usable, efficient, and productive. HCD is a design process that focuses on creating designs based on information about the people who will be using them (Sensei Greenhouse, 2012). HCD builds on and extends the framework established by user-centered design, which focuses on the human’s goals, preferences, tools needed, and tasks performed, to ensure that the end system will be best suited for what the user needs (Norman & Draper, 1986). HCD extends user-centered design by considering not just the end user but all of the humans who may interact with or be affected by the system. HCD involves applying human factors, ergonomics, and usability heuristics to the system design process. Thus, HCD is a process that is system design oriented (vs. user feedback oriented) and is capable of being applied early in the design process, before there is even a system that could enable user feedback. HCD is particularly relevant for systems with stakeholders, in addition to the end user (e.g., maintainers, supervisors, supported users, system administrators), who interact with the system and for systems where adoption and acceptance are based on mission needs (e.g., military systems, industrial process control systems, corporate financial management systems) rather than user preferences.

Great strides have been achieved in implementing HCD in consumer product development; however, it has been much more challenging to incorporate HCD into large-scale development governed by the systems engineering (SE) process. SE approaches design and development by focusing on the complete system and deconstructing it into its multiple components (International Council on Systems Engineering, 2015). During design, the SE process largely focuses on the technical system, often viewing the human as an external actor. However, a true systems approach would recognize that the human is a vital part of many systems and that the human plays an integral role in determining overall system performance and degree of mission success.
The design of the National Aeronautics and Space Administration’s crew exploration vehicle Orion provides an example of the need to consider the human as an internal component of the system. Not only did the engineering design team need to be concerned with the crew’s safety, mission tasks, and human needs, but the presence of the crew also had a direct impact on subsystem requirements. For example, the total system’s (including the crew’s) mass directly affected the launch vehicle requirements. Launch vehicle limitations encouraged lower mass and volume requirements, whereas the ability for the human crew to perform their duties encouraged larger volume requirements—thus creating a trade-off between human performance and volume, both of which have impacts for overall system and mission performance (Morin et al., 2008).

One approach to increase the visibility of human considerations during system design is to fold the HCD process into the SE process rather than performing the HCD process separately from the SE design process. This paper seeks to demonstrate a method for incorporating human capabilities and needs into system designs, by focusing on accounting for the human within system modeling products using the system modeling language (SysML), a graphical language that provides a means of communicating and visualizing system design information. In SysML, the human is typically represented as an external actor rather than as an internal component of the system (Delligatti, 2014). By folding HCD into traditional SE products, human factors engineers gain a way to incorporate critical human considerations into their existing SE framework, thus improving communication between design disciplines during system design and ideally improving system design.

**BACKGROUND**

**Processes, Methods, and Tools**

Human-centered considerations can be embedded into SE practices at the process, method, or tool level. For the purposes of this paper, a process is defined as the set and order of what activities should be accomplished to achieve an objective. Methods support processes by defining in greater detail how to accomplish those activities. Tools are the enabling mechanisms that facilitate and enhance the implementation of a given method (Martin, 1996). There may be more than one tool capable of supporting a particular method; likewise, there could be multiple methods capable of supporting a process. For example, as part of an overall SE process for architectural and logic design activities, a growing method is the object-oriented systems engineering methodology that defines a sequenced set of activities for operational and system design. This method is implemented with SysML, which is implemented in a software-based design tool.

**SE Process, Methods, and Tools**

SE is a process that has become an increasingly important part of the overall life cycle management of Department of Defense (DoD) systems, to the point of becoming an institutionalized disciplinary approach to the development of defense acquisition programs (DoD, 2015). To support this consideration throughout the life cycle, the SE process is composed of 22 technical and technical management subprocesses, ranging from stakeholder needs and requirements definition to architecture definition, integration, verification, and disposal (International Council on Systems Engineering, 2015).

Traditionally, SE has been implemented through a document-based method where design is driven by the development of a set of documents, such as requirements or design specifications. This document-based approach has begun to be replaced by a model-driven approach: model-based SE (MBSE). MBSE allows for the development of the same information through a series of tightly integrated and interrelated models that form a complete system model (Friedenthal, Moore, & Steiner, 2014). The MBSE method results in improved team communication, increased quality of the system’s specification and design, and the ability to reuse the model throughout the system’s life cycle (Friedenthal et al., 2014).

If MBSE is a method of practicing SE, then SysML can be considered a tool with which to implement MBSE. There have been many
graphical modeling languages available for SE applications over the years, and SysML is just one of the latest (Delligatti, 2014). Others include the Integrated Definition family, functional flow block diagrams, UML (Unified Modeling Language), and data flow diagrams. SysML (an extension of UML) is a graphical language that provides a means of communicating and visualizing system design information via a selection of nine uniquely purposed diagrams. These diagrams are composed of a standardized set of precise, unambiguous graphical notations, enabling the modeler to communicate requirements and behavioral, functional, and structural aspects of the system. Figure 1 displays the diagram types and their relationships to one another. SysML provides a language to describe a system; however, it does not provide a modeling methodology (Delligatti, 2014).

**Human Systems Integration**

The human should be a critical consideration during system development and the SE process in general. The U.S. Air Force seeks to extend human considerations beyond human factors (designing products, systems, and processes to account for the human and the interaction of these products, systems, and processes with the human) by establishing a process to trade off all aspects that involve humans, from traditional human factors engineering (HFE) to broader considerations beyond the system design, such as manpower and training. The U.S. Air Force (2010) defines human systems integration (HSI) as the “process by which to design and develop systems that effectively and affordably integrate human capabilities and limitations.” HSI is the process of integrating—and making trade-offs of—human considerations across nine domains: HFE, manpower, personnel, training, survivability, habitability, environment, safety, and occupational health.

There are various methods of practicing HFE—many of which have aspects that easily relate to and align with SE methods—including mission and scenario analysis, function analysis, function allocation, and task analysis. Although it is useful for human factors engineers to use their own set of tools that support these HFE methods, maintaining a separate set of design products provides a potential barrier to team communication. HFEs can build on their knowledge of domain-specific tools, such as timelines, function flow diagrams, and operational sequence diagrams (NATO Defence Research Group Panel 8, 1999), to aid in communication of human-centric needs, requirements, and limitations.

There have been several efforts to better integrate human systems considerations into the SE process. These efforts strive to solve the problem by focusing on the issue at varying levels of scope: the process level, the methods level, and the tools level.

**Process-Level Integration**

Integration efforts at the process level strive to fundamentally change or augment the SE and/or HSI process. Chua and Feigh (2011)
offer various ways in which human factors may be generally included in early system development. They organize their ideas according to four system design stages: requirements acquisition, concept generation, preliminary, and detailed design. Chua and Feigh provide, admittedly at a high level of detail, general suggestions in an effort to encourage communication between systems engineers and human factors engineers and to promote awareness of human factors during system design.

Hardman and Colombi (2012) extend the idea of augmenting the SE process by highlighting the necessity for quantitative methods of expressing HSI requirements, permitting these requirements to be properly considered by program management during system development. As such, these authors outline areas in which to emphasize HSI throughout early requirements analysis, function allocation, and design, further suggesting the usage of empirical data, such as safety investigations and human-subjects experiments, to minimize subjectivity.

Another process-level idea is to standardize the terminology between SE and HSI practitioners. Hardman, Colombi, Jacques, and Miller (2008) suggest clarifications to terminology used across the DoD to reduce inconsistencies among numerous DoD, SE, and human factors and HSI publications: the DoD architecture framework (DoDAF; DoD, 2009), the defense acquisition guide (DoD, 2013), and the International Council on Systems Engineering’s (2015) SE handbook. The idea of standardization may be extended from the DoD to the entire SE community (Madni, 2009; Orellana & Madni, 2014). Orellana and Madni (2014) argue that there is a lack of HSI efforts because of differences in terminology. A proposed solution is to build a common ontology to connect the semantics of the two fields, thus providing a means to address HSI concerns during system design (Madni, 2009; Orellana & Madni, 2014). Bruseberg (2008) corroborates Orellana and Madni’s claim, citing several examples of differences between HSI and SE in their interpretations of terminology. For instance, whereas the term activity has a high-level operational connotation to systems engineers, its scope is often of a low-level human task to human factors engineers.

Although these process-level integration efforts take strides toward increased communication and merging of HFE and SE, they do not guarantee—or even provide a mechanism—to directly incorporate HCD considerations into SE products.

Methods-Level Integration

Efforts at the methods level strive to enhance integration of SE and HFE by seeking to improve one of the existing SE design or analysis methods or proposing a new method. Crisp, Hoang, Karangelen, and Britton (2000) do the latter. Continuing the ideas put forth by Hardman et al. (2008), Orellana and Madni (2014), and Bruseberg (2008), once a common language between SE and HFE is established, Crisp et al. (2000) propose a way to further establish effective integration through an integrated information data repository featuring a common data interchange format. This repository responds to the need for systems engineers to synchronize and balance multiple disciplines, providing a software interchange and physical exchange standard that could implement a common data schema to translate information among disciplines’ unique software tools.

Hardman et al. (2008) and Piaszczyk (2011) propose an augmentation to the DoDAF to improve integration. They examine how each of the nine HSI domains are related to the existing 51 DoDAF views for capturing design information. For example, since the manpower domains deal with the numbers of users, operators, and maintainers, this design information could be captured by DoDAF’s operational views to identify and describe performers, organizations, or organizational types. HFE is a key domain to address system limitations as a result of human involvement and human system interaction. Several DoDAF views could be used to identify risk areas or trade-off opportunities, especially across the human-computer interface, such as the systems interface description (SV-1), systems-systems matrix (SV-3), and the systems functionality description (SV-4). The methods proposed by Hardman et al. (2008) and Piaszczyk (2011) present ways to include HSI in the DoDAF without developing new products.

An alternative integration method is to create new, human-focused views to augment existing
architecture frameworks. In 2007, representatives from the United States, United Kingdom, Canada, and the Netherlands convened the North Atlantic Treaty Organization (NATO) Human View Panel to examine the current state of human modeling descriptions. They proposed a standard human viewpoint that could be adopted by any architecture framework (Handley & Smillie, 2008). The resultant NATO human viewpoint is composed of eight products:

- HV-A: Concept
- HV-B: Constraints
- HV-C: Tasks
- HV-D: Roles
- HV-E: Human network
- HV-F: Training
- HV-G: Metrics
- HV-H: Human dynamics

All these products are designed to address different human aspects that are important to consider during system design and development. For example, HV-A (concept) offers a high-level look at the human component of the system, whereas HV-B (constraints) focuses on capabilities and limitations that the human brings that affect the system. HV-B can be further subdivided into subviews, such as manpower projection constraints and personnel policy constraints. Since most of these views are static by nature, HV-H (human dynamics) is designed to address the dynamic aspects from each of the other views, to include state changes, conditions, time units, and performance measures. The human view was intended to force systems architects to consider the human in its own architecture framework view instead of arbitrarily adding human considerations into other views. Interestingly, at about the same time, Bruseberg (2008) proposed a human view specifically for the British Ministry of Defence architecture framework (MODAF). Listing several of the same human-related shortcomings in the MODAF, as does Handley and Knapp (2014) for the DoDAF, Bruseberg details ways in which her human view can improve the MODAF’s representation of the human during system development. She argues that human views aid in modeling the “soft systems” human side of system development, thus bridging the communication gap between systems engineers and human factors engineers. The MODAF human view is composed of seven products, HV-A through HV-G. These products largely parallel the NATO human view’s eight products. Another goal of adding a human view directly into an architecture framework is to enable systems engineers and HSI analysts to collaborate early in system development, thus contributing more effectively to design (Smillie & Handley, 2009). Several applications of human views have been described by Handley (2011), Handley and Knapp (2014), and Sharples (2014). However, by creating separate views, these efforts fail to ensure that the system-centric views will account for the human. Although these methods provide a means of capturing human-centered considerations in the system architecture, they do not ensure that human-centered considerations will continue to be captured in the system preliminary or detailed design.

**Tools-Level Integration**

The most in-depth, narrowly scoped way to integrate the HSI and SE processes is to approach integration at a tools level. Efforts at this level focus on improving the way in which tools such as SysML can be used to incorporate the human into SE. Although some researchers advocate the use of modeling and simulation in general to consider HSI (Boy & Narkevicius, 2013), other efforts have used MBSE modeling to accomplish this task. Bodenhamer (2012) provides an excellent start for including human considerations into system-level products, such as those developed with SysML. Bodenhamer states that to understand the human’s interaction with the system, the human must first be deconstructed into the functional components necessary to operate the system. These components include sensory channels, cognitive processing, psychomotor capabilities, and physical interfaces. The system itself must also be deconstructed into its components, treating the user as one of these components. Using a landmine detector system as a case study, Bodenhamer created a high-level architectural concept of the system to demonstrate this concept. He modeled the behavioral aspects of the system by creating
activity and sequence diagrams. Demonstrating that the original products consider the human as an outside actor, Bodenhamer updates the diagrams to include the human as internal interfaces to the system. These diagrams are thus able to visually highlight the human-system interaction that is necessary for mission success. By doing so, Bodenhamer claims that the modeler can identify HSI-related problems that could affect system performance or mission success.

Ramos, Ferreira, and Barcelo (2013) address human integration from the process, methods, and tools levels. As part of their larger effort to enhance the overall SE process, they amalgamate aspects from a variety of methodologies to present a revised, more agile MBSE methodology. However, their main focus is at the tools level. HSI is considered a part of the overall methodology, in which Ramos et al. advocate a systems engineer-focused implementation of HSI via SysML diagrams such as activity and internal block diagrams.

Orellana and Madni (2014) also address integration from multiple levels of scope. After proposing their process-level HSI ontology, they narrow to the tools level. Orellana and Madni’s ontology is influenced by defining the human in terms of SysML diagrams. The goal of the ontology is to “bridge the gap” between systems engineers and human factors engineers by allowing systems engineers to define the human using their own MBSE modeling methods. Orellana and Madni provide a high-level description of ways in which the human can generally be represented through SysML diagrams. Ahram and Karwowski (2009) also recommend a common language by incorporating a HSI framework into systems engineers’ SysML modeling practices.

Research Gap

There have been several efforts to integrate the HSI and SE processes. These efforts have addressed the integration problem from various standpoints: the process level, methods level, and tools level. Numerous processes and methods have been proposed, but relatively few efforts integrate SE and HSI at the tools level. The tools level is where a number of specific design decisions are made; thus, this is the level where decisions have identifiable and measurable implications for human-system interaction. Additionally, most efforts have focused on integration at only the early conceptual or architectural phase of a system’s life cycle. During these early phases, HCD is largely focused on requirements specification. Few integration efforts have focused on the preliminary and detailed design phases, where in-depth analysis of human capabilities and limitations through modeling or usability studies is most likely to occur.

The purpose of this paper is to present a different integration approach by focusing on the tools level used during the preliminary or detailed design phases, later in the system’s life cycle. Ideally, this approach should be used in combination with the other integration efforts so that the human is considered at each phase. By focusing on integration at the tools level during these design phases, the resulting system models allow for human consideration at a lower level of system detail. Figure 2 shows that this paper’s research lies in the preliminary design and detailed design stages of the SE Vee model, in contrast to other integration efforts from the literature, which have been primarily focused on conceptual design.

This paper examines human and system considerations through SysML. These diagrams are sequentially built embracing concepts from both SE and HFE. The premise is that by incorporating HCD directly into SE products—with SysML and/or any integrated analysis tools—systems and human factors engineers will be able to better identify, communicate, and correct design issues or invalid assumptions, thus improving system design.

METHODOLOGY

The methodology of incorporating HCD directly into SE products can serve a number of purposes, including system evaluation, system redesign, and system documentation. The first step in the methodology is selecting a specific system and purpose for integrating HCD into the SE products. Next, the sample system is analyzed to identify the task and information flow. Then, system- and human-centered diagrams are separately created to represent different
viewpoints of the system. These diagrams are compared and analyzed, and new diagrams are created that incorporate system and human considerations into one concordant representation of the system model.

**Case Study Description**

This study uses a synthetic task environment called Vigilant Spirit as the system case scenario with which to demonstrate integration. Vigilant Spirit is used by the U.S. Air Force, 711th Human Performance Wing (HPW) at Wright-Patterson Air Force Base, Ohio, to conduct human-in-the-loop experiments studying the effects of certain tasks on participants’ performance, workload, and physiology (Hoepf, Middendorf, Epling, & Galster, 2015). This system is a research prototype that will be modified, many times over, as new research investigations are developed. Thus, this paper demonstrates the creation of an “as is” baseline to support future “to be” modifications. In particular, this analysis focuses on how aspects relevant to operator workload and human/system allocation affect interface design, because these areas are the focus of many research studies conducted with Vigilant Spirit.

Vigilant Spirit was designed to simulate remotely piloted aircraft (RPA) missions, with a single operator controlling multiple RPAs. The multiaircraft control mission contains many interesting human-system challenges (Colombi et al., 2012). This synthetic task environment is a suitable case study because the overall system requires a combination of human and system activities. Vigilant Spirit is a relatively simple system, allowing the study to remain narrow in scope to focus on the methodology instead of the intricate details of a complex system. This scope allows for results to be more easily generalized to other systems.

Using Vigilant Spirit, the pilot performs a surveillance mission attempting to locate and track a high-value target (HVT) who is walking through an urban marketplace. The HVT is identified by the specific weapon (rifle) it is carrying, but there are also distractors walking throughout the market who are unarmed, armed with the incorrect weapon (pistol), or carrying shovels. Participants are seated in a control station that displays the simulated RPA camera feed and a communications window. They use a computer mouse to click within the camera feed window to move the camera and recenter the RPA’s loiter circle, and they scroll the mouse wheel to zoom the camera in and out. Subjects use a keyboard to indicate to the system when they locate the HVT. Beside the primary task of surveilling the HVT, there is also a secondary communication task. Throughout the mission, the operator is asked a series of math-based route navigation questions through a headset. To answer the questions, the operator uses the keyboard to open a communication line to orally respond via headset. Figure 3 shows the experimental setup for the Vigilant Spirit environment.

**Develop Task and Information Flow**

After the system to be analyzed is identified, the task and information flow are captured to identify the relevant processes and activities, to enable these elements to be accurately represented in subsequent models. The mechanism for capturing the task and information flow will depend on the specific system’s design maturity. For example, if the system is in conceptual or preliminary design, this step will be performed by analyzing the functional architecture. However, if an “as is” system exists (or at least a prototype), then a...
task analysis would be appropriate. Additionally, because the human is now inside the boundary of the system, identification of task and information flows may include tasks required of the operator to perform the mission that are outside the boundary of the system under design.

This study’s task analysis of Vigilant Spirit is accomplished through physical observation of the simulation itself during an experimental dry run, as well as through analysis of the human subjects’ data collected by the 711th HPW. The behavioral data set from the 711th HPW’s experiment is used to analyze the activities that the subjects accomplished, the order that activities were performed, and tasks in which the subjects succeeded or failed. The analysis is used to build task networks as a way of visually representing system and human tasks.

**Build System-Centered Diagrams**

In some cases, the SysML diagrams may already exist. If not, the systems engineer will need to build these diagrams. To build the SysML diagrams of the system, the necessary information must first be identified and collected. The requisite information is dictated to some extent by the focus of the modeler, whether looking at structural, physical, or behavioral aspects of the system. Regardless of focus, at a minimum the information collected will include identification of relevant subsystems, how they communicate with one another and with external entities, and what information is passed back and forth therein. The task analysis’s identification of the system’s internal tasks and processes is particularly useful for building activity or sequence diagrams because the focus of such diagrams is to represent the activities involved in performing a certain mission, with varying levels of detail. Within this study’s human context, the most relevant diagrams will be behavior and activity based.

**Build Human-Centered Diagrams**

Building the human-centered diagrams is accomplished similarly to the system-centered diagrams. Whereas the system-centered diagrams represent the system primarily from the systems engineer’s point of view, the human-centered diagrams instead represent that same system from the unique perspective of the human factors engineer. Building these human-centered diagrams can be accomplished by using an HCD approach—focusing on how the human interacts with the other subsystems. The specific human-centered content that is captured in these diagrams will depend on the goals of the current effort and those factors deemed to be most relevant by the human systems integrator. Examples of operator-based aspects to consider for potential inclusion when developing these diagrams include the following: workload, task-load, attention availability or capacity, cognitive channels, physical limitations, physical or
ergonomic requirements, manpower, learning, errors, training, task flow, knowledge, skills, abilities, and communication.

Although using HCD does not create requirements to include or exclude specific items from any diagrams, each SysML diagram is tailored to convey specific information about the system; thus, the selection of which diagrams are created with human-centered versions implicitly specifies the type of data that will be included. For example, sequence diagrams would require the specification of interfaces and data flows between the human and other subsystems and the sequence of those data flows, whereas activity diagrams will capture decision-making processes and decision flows. Depending on the goals of the analysis, other items may need to be included in the diagram, which might require additional SysML diagrams suited to those aspects of the system.

If possible, human factors engineers should interact directly with the end user or other stakeholders, to define human consideration based on how the system is experienced from the user’s perspective. Through user feedback, aspects of the human and its interaction with the system may be uncovered that would otherwise go unaccounted.

In the case of developing the Vigilant Spirit “as is” baseline, the purpose in creating the human-centered diagrams is to effectively communicate with engineering design team; thus, we focus on system interface and how the user communicates with the system. What interfaces are used to communicate with the system, and which of the user’s senses are utilized to interact with those interfaces? Cognitive processes are also analyzed with respect to these interactions: the choices or decisions that the user makes, how the interface design affects the user’s workflow, and the user’s desired workflow. The focus for these diagrams is on the user’s interaction with the system. As part of the HCD approach, the process of understanding and modeling the system may involve a few iterations of user or stakeholder interaction to get the diagrams to a desired level of design maturity.

**Compare and Analyze the Differences**

The generated system- and human-centered diagrams are qualitatively compared to identify and analyze the differences between them. When comparing diagrams, systems engineers and human factors engineers are seeking to identify (1) unique information that needs to be captured in the integrated diagrams and (2) common anchor points that can be used to link this information. The information that needs to be captured in the integrated diagrams will depend on the specific purpose for which these diagrams are being generated. In general, the human factors engineer should indicate the system design requirements that have emerged from creating the human-centered diagrams. In this case example, the goal is to communicate with the engineering design team to support future redesign efforts; thus, the focus is on the system interface. Specifically, the concern is with how operator workload and cognitive channels affect interface design; thus, the human factors engineer needs to identify system requirements relating to these aspects. However, if we were focusing on a different aspect (e.g., training), then different features would be retained for integration (e.g., information on tasks performed by the human operator).

To identify the common anchor points, the relevant diagrams from both the system’s and the human’s focus are examined for similar elements. These common elements found in both sets of diagrams may be used as common anchor points with which to compare the system’s handling of tasks versus that of the human. For example, a single task may include both human and system involvement; therefore, the task will appear on each of the separate diagrams. This common task would serve as an anchor point, connecting the separate human and system inputs that feed into that task. Since this case example is based on SysML activity and sequence diagrams, the anchor points can include any of the typical diagram elements:

- Tasks, functions, activities
- Data and information objects
- Interfaces—shown as an object or control passing between subsystems, where each subsystem is indicated in a unique swim lane to improve interpretability

**Create an Integrated Set of Diagrams**

The key differences noted by comparing the separate system- and human-centered diagrams can be used to create new diagrams that integrate
the system and human perspectives. The information that should be included in the integrated diagrams will depend on the specific goals of the systems and human factors engineers. Any system design requirements that have emerged from creating the human-centered diagrams should be captured in the integrated diagrams. The result of creating an integrated set of diagrams is a single set of depictions that account for both the system and the human—enabling the system designers to perform trade studies that account for human-system interactions and human performance. Reiteration with the same HCD concepts as in the human-centered diagrams may also be helpful with these integrated diagrams to ensure that relevant human considerations have been maintained.

Limitations and Assumptions

To apply the method described herein, several assumptions are made, including the establishment of initial performance and functional requirements, manpower requirements, allocation of functions to humans and systems, and preliminary design. These aspects of the system are necessary to effectively generate the system- and human-centered SysML diagrams. These items need to exist in only their preliminary form; it is quite possible that these initial requirements, allocations, and designs may be altered after the creation of the human-centered diagrams and the incorporation of emergent requirements into the integrated diagrams.

One limitation of this method is that it considers only one design implementation at a time. Thus, if the system and human factors engineers are evaluating multiple design options, each of these design options will require its own iteration. Another limitation is that this analysis is qualitative/logical and not quantitative. However, the diagrams produced through this analysis can be used to inform quantitative analyses.

RESULTS AND ANALYSIS

The results of the “develop task and information flows” step revealed three separate processes occurring during the simulation: two system processes and one human process. The system has an independent set of activities that it performs for the surveillance and communication tasks, each of which precipitates response activities from the human operator. For example, the system spawns a random HVT at the beginning of each of four iterations, for which the operator must search, indicate if found, and then zoom in and follow. The system also asks four iterations of communications questions, prompting calculations and answers from the operator. A SysML activity diagram was selected to represent these tasks, as shown in Figure 4. Activity diagrams are conducive to representing task flows during early system design because they are able to visually depict mission activities at a high level, allowing modelers to consider the actors (vertical swim lanes), decisions, and task flows involved.

The activity diagram in Figure 4 also offered our first look at representing the system through SysML diagrams. Its broad depiction of the system’s activities and interactions served as a basis with which to expand on and incorporate more details in new diagrams. Sequence diagrams were used for this purpose, as they are better suited for illustration of subsystem activities and intersystem communication.

The Vigilant Spirit system is composed of two subsystems: surveillance and communication. The surveillance and communication tasks occur independently of each other from a system standpoint and would likely use different hardware, which may be obtained in separate acquisition programs in a real-world implementation. Therefore, we divided the surveillance and communication subsystems into their individual components, with separate sequence diagrams, instead of representing them as one system. Doing so allows for a functional allocation of who or what will be handling these different system aspects. The system-focused sequence diagrams of the surveillance and communication tasks are shown in Figures 5 and 6, respectively. These make use of a generalized model-view-controller software design pattern internally. In the figures, the human is labeled an external “actor.” This is the traditional software engineering method of representing the human as an external (from system) entity with goals and roles.
In the surveillance diagram in Figure 5, the system is divided into five abstract subcomponents: the user interface (UI), controller, target, distractors, and score. The use of a UI and controller is common when depicting software-based systems. Note that even though these are system-centered diagrams, the human is still represented to a degree. The operator, represented by a single lifeline on the leftmost side of the diagram, interacts solely with the UI. The controller manages the system’s activities and timing, creates and manipulates objects (e.g., the HVT and distractors), and delegates tasks, such as continuously updating the score until the mission has ended. The sequence diagram generally depicts the same task flow as the activity diagram but with more details of what data or information is consumed or generated with each task.

For example, searching for the HVT consists of the operator continuously sending commands to the controller via the UI to move the RPA camera and adjust the zoom level until the operator finds the HVT. By contrast, the activity diagram in Figure 4 represents searching for the HVT simply by a single action node. The communication diagram in Figure 6 is similarly focused, with the system divided into four subcomponents: the UI, controller, question bank, and score.

Because the sequence diagrams were built from a systems engineer’s perspective, less emphasis is placed on the (external) user. These are now generated from a human factors engineer’s perspective. The construction of the human-centered diagrams can be the creation of completely separate diagrams or additions, deletions, or decompositions of the existing SE diagrams. In this case, the objective is to detail the modalities in which the user interacts with the system; thus, the diagrams are modified to con-

Figure 4. Activity diagram of Vigilant Spirit tasks. HVT = high-value target.
Figure 5. Sequence diagram of system-centered surveillance task. HVT = high-value target; UI = user interface.

Figure 6. Sequence diagram of system-centered communication task. UI = user interface.
vey this information by decomposing the human element. In the same manner that the system was split into subcomponents, the user can be represented by several resources (Wickens, 2008), where each performs specific tasks. For example, listening tasks, such as hearing the communication question, can be performed by only the human’s auditory system. Likewise, response tasks, such as indicating the HVT as found and answering the question, are performed by the human’s motor systems. Although Vigilant Spirit is an existing system, if the system had not yet been designed, the human factors engineer would have options for the implementation of certain tasks. For example, responding to the communication question could occur orally through headset or manually through keyboard. These options could have implications not only for system design but for overall system performance. Note that the system-centric diagrams focus on function allocation but do not account for the level of design specification necessary for human factors engineers to assess human performance implications.

As these diagrams are purely human centered, less emphasis is placed on intersystem events, and instead the system is abstracted to just focus on its interaction with the user. The redesigned, human-centered sequence diagrams are shown in Figures 7 and 8, in which the human is divided into its visual, auditory, cognitive, and psychomotor components (denoted “human”), and the system’s UI is further divided into three subcomponents with which the user interacts: the computer keyboard, mouse, and monitor (denoted as “UI”).

Having sets of diagrams from the system’s and human’s perspectives, we qualitatively compared the diagrams to find similarities and differences. The system-centered sequence diagrams contained detailed depictions of Vigilant Spirit’s sub-systems and its intersystem communication while treating the user as a “black box.” Conversely, the human-centered sequence diagrams focused on the human’s resources and interfaces with the system while treating the system as a “black box.” However, each set of diagrams’ narrow focus is also its unique strength, providing system and human insights into Vigilant Spirit that demonstrate the benefit of creating an integrated set. Because the medium with which the user and system communicate to each other is the UI, this served as the bridge to connect the two diagram perspectives. The integrated diagrams are shown in Figures 9 and 10, with the UI’s subcomponents denoted as “UI.”

Because the goal of creating the integrated set of diagrams is to convey interface design information to the engineering design team, incorporating the UI’s subcomponents into the integrated diagrams provides the systems engineer insight into human-system interaction without needing to include the human’s resources. Thus, the diagrams allow for the necessary amount of detail for a systems engineer’s area of interest in a modeling language with which the engineer is familiar. It is important that the human is considered and included in the system diagrams; its resources are implied by the UI breakout and sufficiently represented therein.

The benefit of creating these integrated diagrams is the ability to gain insight into the human processes and interfaces involved while the user is interacting with Vigilant Spirit. For instance, by depicting the user as being internal to the system and by capturing the types of interfaces with which the user interacts and when the user must use them, the potential for imbalance of resource allocation may be more easily identified. An example of an imbalance would be if the user were required to answer the question by typing the answer while still needing to search for and indicate the HVT, thus requiring the use of the same keyboard and mouse interface for concurrent tasks. When the human is considered external to the system, these conflicting demands on the user’s limited capabilities—expecting the same user to accomplish two tasks simultaneously—would not be apparent, whereas they are easily realized when the human is internal to the system. This benefit of identifying human capabilities and limitations comes without needing to sacrifice detailed models of Vigilant Spirit and its subsystems.

Another important aspect of MBSE is the power to now integrate these early design definitions (Figures 5–10) with analysis. SysML includes a parametric diagram. This is used to define and relate constraints, metrics, and design parameters of system components (i.e., blocks). Many tools allow these parameters to be simulated
outside a descriptive design tool. This creates the potential to experience the results of design decisions by

- conceptually designing a human-system in SysML;
- importing those diagrams into a MBSE tool that enables dynamic simulation of the models;
- assessing the effectiveness of select design parameters;
- conducting trade studies or sensitivity analysis by adjusting those design parameters; and, finally,

Figure 7. Sequence diagram of human-centered surveillance task. HVT = high value target; Op = operator; UI = user interface.

Figure 8. Sequence diagram of human-centered communication task. Op = operator; UI = user interface.
optimizing the selected design parameters to achieve high performance.

In our case example, potential upgrades to Vigilant Spirit include a number of system redesign options that involve making trade-offs—for example, camera feed quality (resolution) versus video-feed delay time or automated searching algorithms speed versus accuracy. These trade-offs, including the impact of human performance, could be simulated to select the parameters that would achieve the highest overall system performance.

DISCUSSION AND CONCLUSION

Aside from the specific Vigilant Spirit scenario, creating a set of SysML diagrams embracing HCD concepts provides advantages for system designs in other contexts. Systems engineers understand the benefit of dividing the system into its subcomponents, functionally decomposing system requirements, apportioning constraints and -ilities (reliability, availability, maintainability), and balancing cost, schedule, risk, and performance. Human-centered diagrams, based on SysML, represented the human’s visual, auditory, cognitive, and psychomotor subcomponents as resources and its use of system interfaces. By revising system diagrams to include the human, systems engineers can gain insight into (1) the possibility for the human to perform all or some of its allocated tasks; (2) the potential for conflicts, workload issues, and changing interface designs;

Figure 9. Sequence diagram of integrated surveillance task. HVT = high-value target; UI = user interface.
and (3) the use of targeted autonomy. These considerations are not necessarily accounted for by normal SE practices.

Future work in this area of study will focus on further bolstering human considerations into SE processes with human performance modeling. A discrete event simulation software tool, IMPRINT (Improved Performance Research Integration Tool), will be used to capture the dynamic aspects of the human’s cognitive performance. By using IMPRINT, it is possible to analyze the interaction between the system and human across a range of dynamic activities occurring in the Vigilant Spirit environment. Additionally, this proposed integration method need not be limited to the specific MBSE and HSI tools that this research used. There may be other modeling tools besides SysML that would yield the same benefits from integrating human considerations. Likewise, although this paper focused on the HFE domain to integrate, a future goal is to expand integration efforts across the rest of the nine HSI domains.

To better integrate human considerations into system designs, it is necessary for systems engineers to first acknowledge and consider the human as an important part of the system. However, mere acknowledgment is not enough if the human is not sufficiently integrated into the SE process. Similarly, human factors engineers need to be a part of the SE process to ensure sufficient human integration. Integration needs to be sufficiently scoped and at a level of detail that is able to capture the important aspects of the human (e.g., workload, taskload, attention availability or capacity, cognitive channels, physical limitations, physical or ergonomic requirements, manpower, learning, errors, training, task flow, knowledge, skills, abilities, and communication) as well as implications for human-based system performance effects. At the tools level, SysML is a language that enables the integration of human considerations into SE products, and this study’s approach provides an avenue to achieve that goal. The details of this integration—with respect to which diagrams to modify and what

Figure 10. Sequence diagram of integrated communication task. UI = user interface.
human-specific information needs to be included—will ultimately depend on the specific purpose for which the diagrams are being generated. Incorporating HCD early in the SE process will allow for a reduction in total system life cycle cost, while still achieving human-system effectiveness—the mandate of HSI.

ACKNOWLEDGMENTS

This work was supported in part by the Human Research and Engineering Directorate, U.S. Army Research Laboratory. The views expressed in this paper are those of the authors and do not reflect the official policy or position of the U.S. Air Force, the U.S. Army, the Department of Defense, or the U.S. government.

REFERENCES


Michael E. Watson received a bachelor of science in aerospace engineering from The Pennsylvania State University in 2009. He recently graduated with a master of science in systems engineering from the Air Force Institute of Technology in 2016, where his master’s research focused on human systems integration. Capt. Watson is currently a project engineer at the Space and Missile Systems Center at Los Angeles Air Force Base, California.

Christina F. Rusnock is an assistant professor of systems engineering at the Air Force Institute of Technology. She earned her PhD in industrial engineering in 2013, a MS degree in industrial engineering in 2011, a MS degree in R&D management in 2008, and a BA in economics-government in 2004. Her research interests include human performance modeling, mental workload, and situation awareness, with a focus on human-computer interaction. She is a member of the Institute of Industrial Engineers and the Human Factors and Ergonomics Society.

John M. Colombi is an associate professor of systems engineering at the Air Force Institute of Technology. He received his PhD in 1996 and MSEE in 1992 from the Air Force Institute of Technology, with a BSEE degree from the University of Lowell. He has 21 years of military experience as a developmental engineer. His research interests include systems and enterprise architecture, complex adaptive systems, acquisition process modeling, and human systems integration. He is a member of the International Council on Systems Engineering and IEEE.

Michael E. Miller is an associate professor of human systems integration at the Air Force Institute of Technology. His research interests include human systems integration, modeling and measurement of human performance, and human interface design, particularly human-display integration. Prior to joining the Air Force Institute of Technology, Dr. Miller spent more than 15 years with Eastman Kodak Company as a systems and human factors engineer. He is a member of the International Council on Systems Engineering, Human Factors and Ergonomics Society, and a senior member of the Human Factors and Ergonomics Society, and a senior member of the Society for Information Display.