Boy, G.A. (2021). Human Systems Integration and Design, Chapter 2 of the *Handbook of Human Factors and Ergonomics*, Gavriel Salvendy & W. Karwowski (Eds.), 5th edition, Wiley, USA.

CHAPTER 2

HUMAN SYSTEMS INTEGRATION AND DESIGN

Guy A. Boy
CentraleSupélec
Paris Saclay University
Gif-sur-Yvette, France

ESTIA Institute of Technology
Bidart, France

1 INTRODUCTION

Human Systems Integration (HSI) denotes processes and results of contemporary systems engineering that concurrently considers technological, organizational, and human factors during a system's entire life cycle. This life cycle includes design, development, certification, delivery, operation, and decommissioning. The concept of "system" represents people, organizations, and machines, who are cognitively and physically defined in terms of structures and functions. The system's physical and cognitive attributes can be designed to satisfy a set of principles and criteria, which lead to a set of requirements in terms of tasks and activities. After reviewing the evolution of human-centered approaches, this chapter delves into the overall question of system from human-centered system science and development points of view. We propose a foundation for HSI and human-centered design (HCD), with examples from aerospace and more generally life-critical industrial examples to illustrate theoretical developments.

2 SOME HISTORY AND DEFINITIONS

Scientists labor over definitions and delineations of concepts in their quest to improve their various meanings in context. The meaning of a concept may vary with respect to culture, and general usage. This section contextualizes human systems integration (HSI) in its historical landscape.

2.1 Human Factors and Ergonomics

Human factors and ergonomics (HFE) was the first and most dominant field of investigation that seriously considered people's issues at work (Tillman, Fitts, Woodson, Rose-Sundholm, & Tillman, 2016). HFE started in the post-World War II era. HFE was uniquely a matter of physical ergonomics until the 1980s. Before the 1980s, engineering was dominated by hardware, led by mechanical engineers. HFE was dominated by physicians (i.e., workplace medical doctors) dealing with health and safety issues at work. These physicians were involved in solving problems relating to physiological and biomechanical issues.

The advent of personal computers, also in the 1980s, totally changed the way people work and live. Information overload was responsible for novel issues at work. Consequently, HFE became more cognitive to better understand and manage the shift from doing to thinking (i.e., jobs and, more generally, activities became more cognitive; physical tasks being phased out to machines). HFE then became a business for psychologists.

For a long time, HFE approaches and methods were primarily corrective with very little prescription. Engineering came, and still comes, first, and HFE comes in when a system is fully developed and ready to be tested. HFE's traditional role is evaluation, with a meager contribution to design and development. The reason is that HFE evaluations require a workable completed system, not only parts of it, to be tested. Consequently, activity analysis is typically performed using existing systems, which imposes a continuity approach to ergonomics (i.e., moving to a new system that is an evolution, and not a revolution, of an existing integrated system). Indeed, HFE is mainly used at the end of the development process to assess the system before delivery. Therefore, if HFE is implicitly concomitant with a continuity-based design approach, what happens when we adopt a disruptive (revolutionary) design approach? HFE analysis of an existing system will very seldom help in the design of a new system, and we need to wait for the system to be completed to use HFE methods to test it.

During the 1990s, HFE scientists and practitioners raised the automation "surprise" issue (i.e., unanticipated effects of automation). What they did not study, however, was the maturity issue. Indeed, maturity is a matter of technology, organization, and people. An immature system requires technological, organizational, and human expertise to be handled correctly. In other words, early-stage system operations demand highly skilled and knowledgeable human operators to handle them correctly (e.g., at the beginning of the twentieth century, cars were driven by chauffeurs, who were not only qualified to drive an automobile, but had the expertise to repair it in case of failure; at the end of the twentieth century, however, almost everyone was driving a car with no expertise in car engines). This is a matter of technological maturity (i.e., toward minimizing technological surprises). At the same time, cars specialized with respect to practice (e.g., family cars, trucks, and racing cars). This is a matter of maturity of practice (i.e., toward minimizing operational surprises).

HFE makes a distinction between a (prescribed) task and an (effective) activity (Kaptelinin & Nardi, 2006). As already stated, HFE activity analysis can be performed either before design starts or after engineering delivers a product, but not during the design and development processes. This chapter provides methods and tools that ensure human and organizational activities are considered seriously in the design and development processes.

2.2 Automation, Human Engineering, and Cognitive Engineering

Automation was massively used to replace several human functions by machine functions, which increased accuracy and decreased operational costs. However, it also introduced unanticipated side-effects, such as higher stress (i.e., excess of cognitive, perceptual, and attentional workload) or lower vigilance (i.e., complacency) in some cases. Several authors have reported issues, such as human errors, automation surprises, new attentional demands (attentional and coordination), and the danger of missing critical events (Billings, 1996; Hollnagel, 1993; Sarter, Woods, & Billings, 1997).

In Section 2.1, we saw that HFE was dominated by life and social sciences professionals. However, research and practice combining in parallel electrical and mechanical engineering began, more specifically in control and automation. Human engineering was thus born, where human operator models were developed using engineering mathematical and conceptual models (Borah, Young, & Curry, 1988; McRuer & Krendel, 1974; Wiener & Nagel, 1988).

Several human performance models (HPMs) have been developed over the years to enable both prediction and explanation of human performance in life-critical situations. Both predictive and explicative models can be used to explore, design, develop, test, and operate life-critical human–machine systems in a non-obtrusive way. For example, MIDAS (Man-Machine Integration Design and Analysis System) is a dynamic and integrated HPM that supports design, visualization, and computational evaluation of complex human–machine systems in simulated operational environments (Corker, 1994; Gore & Jarvis, 2005; Gore et al., 2002; Gore et al., 2009). MIDAS has been developed since 1986 to predict performance in several aerospace domains. MIDAS has been used to evaluate aircraft cockpit technologies for NextGen[1] operations. Transparency and validation remain key issues for these kinds of human operator models, because they are made of interconnected functions that make them almost as complex as real human beings.

At the beginning of the 1980s, HFE and human engineering were evolving from health and safety medicine to experimental psychology. Cognitive engineering was nascent. A technological revolution was growing and Airbus was the first commercial aircraft manufacturer to design and deliver highly-automated cockpits for commercial airplanes. The question was, "how will we certify these new highly-automated machines?" HFE disciplines did not have appropriate concepts and tools to respond effectively. It is interesting to observe that "cognitive engineering" was growing in parallel as a major discipline capable of supporting such an endeavor. Donald Norman was the first to coin the term "cognitive engineering" (Norman, 1986). During this period, many scientists laid down the foundations for cognitive engineering research such as supervisory control (Sheridan, 1987), the skill-rule-knowledge behavioral model (Rasmussen, 1983), joint cognitive systems (Hollnagel & Woods, 2005), organizational interaction (Flores, Graves, Hartfield, & Winograd, 1988), situation actions (Suchman, 1987), distributed cognition (Hutchins, 1995b), and cognitive function analysis (Boy, 1998). Cognitive engineering was born in the growing evolution of cognitive science, cognitive psychology, artificial intelligence, computer science, electrical and computer engineering, and software engineering. Several new disciplines emerged such as human–computer interaction, computer graphics, speech recognition, natural language generation, mobile technologies, ubiquitous computing and, more generally, advanced interaction media.

For example, in the early 1980s, aircraft manufacturers decided to reduce the size of aircrews from three to two in commercial aircraft cockpits. We, aircraft manufacturers and researchers, needed to develop methods and tools for the certification of such new cockpits. Replacing the flight engineer by appropriate systems required a better understanding of the various functions that the captain and the first officer have to handle. Workload appeared to be the main human factors variables to be measured. Models were developed to simulate pilots. For example, MESSAGE[2] was developed for the certification of two-crewmen cockpits, based on a multi-agent HPM model of the aircrew, aircraft and ground systems (Boy & Tessier, 1985). The resulting model helped us to characterize various information densities that were useful in providing an interpretation in terms of workload and performance. MESSAGE workload and performance models have been used in experimental flight tests. MESSAGE supported the definition of appropriate human factors measurements useful for certification purposes.

In this chapter, the term "agent" is used in the artificial intelligence (AI) vein. Marvin Minsky, one of the founders of AI, proposed the definition of an agent as a society of agents (Minsky, 1986). This recursive definition is very useful (i.e., an agent is an agency of smaller-grain agents). We will use the term agent to represent both people and machines equipped with cognitive capabilities. We sometimes talk about cognitive systems to denote agents.

2.3 Human–Computer Interaction

At the same time as cognitive engineering was emerging, Human–Computer Interaction (HCI) was born in the field of computer science. The first ACM CHI[3] conference was held in Gaithersburg, Maryland, USA, in 1982.[4] Since then, CHI conferences have continued to represent the best of HCI research. Office automation was at the center of HCI research and development. Text processing systems were incrementally studied and improved (Card, Moran, & Newell, 1983). Many other computer-based systems have been developed and related uses studied, such as graphical user interfaces and visualization (Shneiderman, 1986, 2002), usability engineering (Nielsen, 1993), participatory design (Bødker, 1996; Grudin, 1993; Muller, 2007), computer-supported cooperative work (CSCW) systems (Poltrock & Grudin, 1994), and other things related to the user experience of computing systems (Buxton, 2010; Norman, 2013). Other conferences were developed, such as the HCI International, providing a large forum for both academia and industry.[5]

Even if cognitive engineering had already been introduced as an evolution of automation and human engineering, there is a strong intersection with HCI, since Donald Norman was fairly involved in automation and HCI when the field really started (Norman, 1986). Today, the field of HCI is developing further in social media and, more generally, in interaction media that are part of our everyday life (e.g., Google, LinkedIn).

From a technical point of view, HCI is at the intersection of artificial intelligence, computer graphics, image recognition, data visualization, and other things (Garcia Belmonte, 2016; Kruchten, 2018). From a human point of view, it is centered on interaction methods, styles, situation awareness, decision making, human–human interaction via computing systems, and interaction with computers using most of our five senses.

One of the main advances in HCI was ubiquitous computing (Weiser, 1991). From a technical point of view, ubiquitous computing consists in connecting electronic devices, including embedding microprocessors to communicate information (e.g., synchronizing your computer with a smartphone at all times, anywhere). From a human point of view, ubiquitous computing enables you to be connected with your data anytime, anywhere, and is also called pervasive computing.

More recently, human–robot interaction (HRI) has developed toward increasingly autonomous robots, which can interact using voice recognition, environment-sensitive sensors (e.g., stereo-cameras), vision systems, tactile interaction, GPS-based tracking systems, and other things.

2.4 Human-Centered Design

Around the turn of the twenty-first century, human-centered design (HCD) became possible when modeling and simulation capabilities were able to provide more realism for human-in-the-loop simulations (HITLS) (Boy, 2013, 2020). The good news was that we could seriously consider human factors during the design process. Consequently, designers not only consider tasks (i.e., what is prescribed to be performed), but also activities (i.e., what is effectively performed using technology being designed and developed).

The HCD of complex systems considers concurrent creation and development of conceptual and/or technological artifacts, together with people and organizations that relate to them. In other words, technology, organizations and people's activities are co-designed and need to be studied using different kinds of scientific methods. We now talk about the TOP model in HCD (Figure 1). The TOP model is typically used in HCD to design and develop technology, taking into account organizational changes, as well as the creation, removal, and transformation of

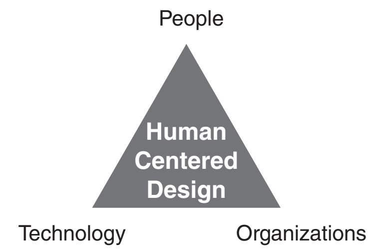


Figure 1 The TOP model. (*Source:* Boy, 2013, 2020).

people's jobs and activities. This model can be used as a starting point from various perspectives, either from a technological innovation perspective, or an organizational change perspective, as well as an evolutionary adaptation or disruptive changes of people's activities.

HCD has been described in several ways. It emerged as a reaction to the rigid world of corporate design and engineering (i.e., current systems engineering), which dictates that engineering leads design and development, and people would be considered when technology was developed by creating user interfaces and operational documentation. Human-centered innovation cannot fit this rigid mold.

In 2005, the Hasso Plattner Institute of Design (known as the d.school) was founded at Stanford University, named after its major donor, SAP co-founder Hasso Plattner, based on the Design Thinking concept developed by David Kelley, Larry Leifer, and Terry Winograd (Kelley, 2016). Design thinking takes "context" into account (i.e., people's requirements, technological possibilities, and economic viability) (Brown, 2008). Design thinking brings flexibility that contrasts with the rigidity of analytical thinking (Plattner, Meinel, & Leifer, 2016). In addition, design thinking incorporates creativity into conventional STEM[6] conventional engineering approaches, promoting a culture of innovation where HCD defines new STEAM[7] approaches. Finally, design thinking deals with change management, as well as organization design and management.

HCD finds its roots in human–computer interaction (HCI), which considers human factors in computing systems and has also become a design discipline. Donald Norman is certainly one of the best promoters of HCD, recognizing the need to observe activity,[8] making a distinction between logic and usage. This leads to the concept of user experience (Edwards & Kasik, 1974; Norman, 1988). HCD encapsulates what Norman (1986) calls user-centered systems design (UCSD). The term "user" may be misleading for two reasons. First, it leads people to think about end users and not necessarily certifiers, maintainers, and trainers, for example. Second, people are more than mere "users," and thus have more characteristics when dealing with systems; they are people!

The NASA Human Systems Integration practitioner's guide provides a very clear and explicit definition of HCD in the space domain (Rochlis Zumbado, 2015). Let's take this as a reference within the TOP model:

- *Concepts of operations (CONOPS) and scenario development (related to everything in the TOP model).* CONOPS define configurations of the overall organization, including people and technology; and scenarios define possible chronologies of the various operations, involving both people and technology within the organization.
- *Task analyses (related to both "P" and "T" of the TOP model).* Task analyses provide prescribed scripts to be performed by both people and technology.

- *Function allocation between humans and systems (related to everything in the TOP model)*. Required functions may be allocated either to people or technology, knowing that each organization can be considered as a system that includes people and technology.
- *Allocation of roles and responsibilities among humans (related to the "P" of the TOP model)*. Following up function allocation, people's roles and responsibilities need to be clearly understood; they are typically based on activity analyses.
- *Iterative conceptual design and prototyping (related to both "T" and "P" of the TOP model)*. We assume that design and development are Agile (Agile Manifesto, 2015), that they are based on prototypes, iteratively refined using human-in-the-loop simulation
- *Empirical testing, e.g., human-in-the-loop testing with a representative population, or model-based assessment of human-system performance (related to the "P" of the TOP model)*. Whether using HITLS or a model-based approach (Kim, Wagner, & Jimenez, 2019), human modeling is needed to support empirical testing.
- *In-situ monitoring of human-system performance during flight (related to both "P" and "T" of the TOP model)*. Human system performance should be measured or assessed to better understand how the overall system works and behaves.

In this chapter, HCD is considered an evolution of engineering, concurrently considering technological, organizational, and human factors. In addition, HCD fundamentals will be presented to better understand and correctly perform when designing complex systems. The AUTOS pyramid was shown to be a very useful support to human-centered design and development of an onboard aircraft weather situation awareness system (Boulnois, 2018), and in the automotive sector (Moertl, Neuhuber, & Pretto, 2019).

Fundamentally, HCD is now possible because we now have HITLS that enables holistic testing from the very beginning of the design process. Activity analyses can be performed, and therefore make agile development more human-centered. In contrast to the past where technological design and development were done from the inside out (i.e., from the kernel of a technology to the user interface), we now can start from a purpose, design and test virtual prototypes, and further tangibilize them (i.e., make them real). Consequently, HCD can be defined as outside in. For example, we can now start from a digital simulation of a physical environment, where all agents in the field are virtual, test orchestrated scenes of all agents interacting with each other and figure out how functions can be better allocated between them. We can start with such a simulated environment displayed in a digital control and management room, and assess the viability of the overall system with real people in this room (i.e., an outside-in approach).

Now that we can start creating and refining systems as pieces of software from the beginning of the design process, tangibility has become a key issue (see Figure 2). We first develop a virtual system that runs HITLS, observing and analyzing human operator activity, and carrying out an agile system development (i.e., performing virtual HCD). Once the structures and functions related to the human-machine system have been defined, a second step can be started, where some components of the systems can become more physically tangible (Letondal et al., 2018), a new agile development can be initiated using the same process as before, and so on. In the third step, the system should be fully tangible, and tested using the same approach as before. This outside-in approach to design and development (i.e., from purpose and usage to core technological means) contrasts with the inside-out traditional engineering approach (i.e., from core technological means to purpose and usage).

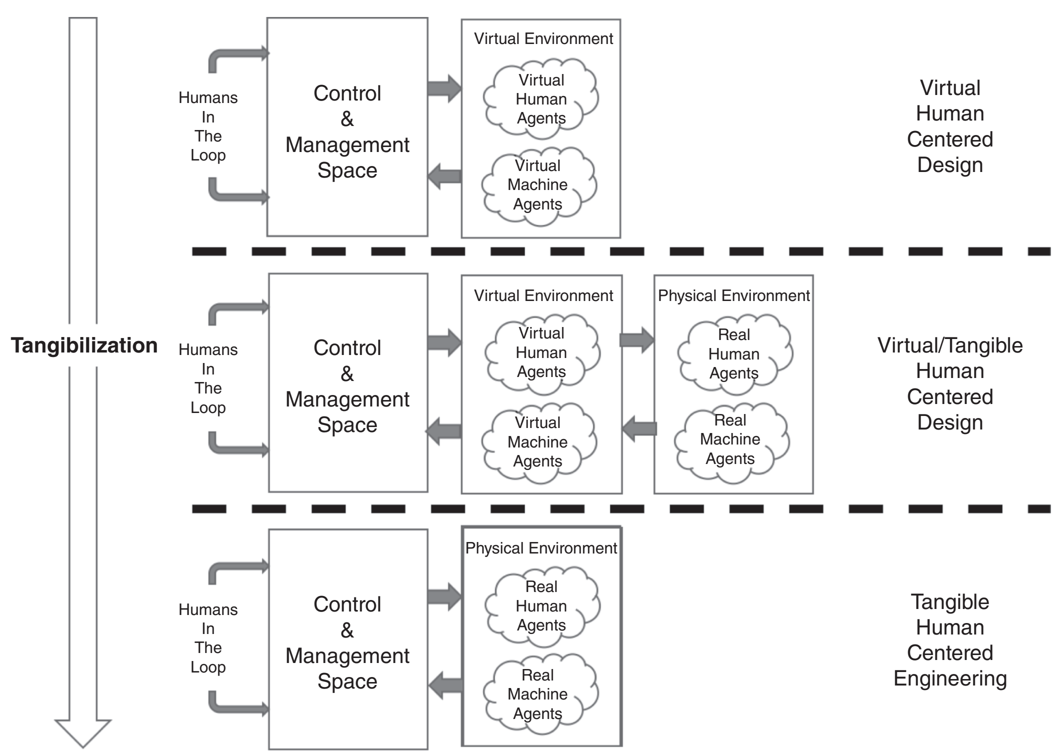


Figure 2 Tangibilization process in three steps: from virtual to tangible.

Humans are now firmly in the loop within a control and management virtual space, which incrementally becomes more tangible. The term "control space" is generic, referring to a control room or a vehicle simulator. Since we deliberately assume that we are in a multi-agent environment, agents being people or machines, we start with virtual agents. These agents do not include the people who are interacting with the control and management space within which we incrementally tangibilize agents in an incrementally more physical environment. For example, let's consider that our goal is the design and development of a fleet of robots replacing people on an oil and gas offshore platform. We will start by developing a control and management room (space) where real people will have to deal with a simulator of both a virtual fleet of robots moving and interacting with a virtual oil and gas offshore platform. We observe the activities of these people and analyze them to produce modifications of structures and functions involved in the simulation. We further pursue this virtual HCD (VHCD) process until a satisfactory design is reached, in terms of safety, efficiency, and comfort, for example. Then, we can start to tangibilize one or two robots, as well as the platform, in a physical playground. A virtual/tangible HCD process can then be initiated as for the VHCD, same agile processes, and so on. The tangibilization process can then continue until everything is tangible (last layer in Figure 2).

2.5 Human Systems Integration

Despite the holistic approach of systems engineering (SE), systems still fail, and sometimes spectacularly so. Constant evolutions make it difficult to stay abreast of requirements, solutions, and environments. SE requires more flexibility, with new approaches being developed that integrate creativity, and where the functions of people and technology are appropriately allocated within our highly interconnected complex organizations. Instead of disregarding complexity as being difficult to handle, we should embrace it, discovering behavioral attractors and the emerging properties that it generates. The concept of agent-environment system "behavioral attractors" has been defined in the context of artificial life (Montebelli, Herrera, & Ziemke, 2007), to denote a parametric space of a systemic topology where an agent tends to stabilize. The parametric space is defined by a set of relevant variables. The search for such parametric spaces, and therefore behavioral attractors, is key to formalizing the concept of familiarity.

HCD provides the creativity factor that SE lacks. HCD is an approach that puts people at the center of the design process of usable and useful systems. Ideation and creativity are crucial processes in HCD, where experienced people in a domain (e.g., pilots in aeronautics) work together with specialists in engineering as well as human and social sciences. Creativity is a matter of integration. For example, when a painter creates a new color, e.g., a new orange, they take a bit of red and a bit of yellow, mix them, integrate them, and then maybe add a little more of one, and so on until a satisfactory orange is found. HCD promotes modeling and simulation from the early stages of design and throughout the life cycle of a product. Combining HCD and SE shapes appropriate human systems integration (HSI) to produce successful systems (Boy & McGovern Narkevicius, 2013). As an example, human space flights are one of the most complex and difficult to handle. NASA has put together an HSI practitioner's guide that is carefully used by NASA personnel (Rochlis Zumbado, 2015). Results show that such support is crucial for successful results. More research is of course needed to demonstrate the various contributions of HSI, in terms of successful systems. The INCOSE[9] HSI Working Group compiles and develops new approaches that address the role of people and organizations in complex systems of systems.

HSI complexity can be analyzed by explaining the emergence of (cognitive) functions that arise from agent activity. This expansion of agents results in a need for function allocation approaches that supersede the classical MABA-MABA[10] model (Fitts, 1951), where functions are not allocated deliberately, but as they emerge from agents' activities. Our socio-technical world is dynamic and the concept of static tasks should also be superseded by the concept of dynamic activities. A task is what we prescribe; an activity is what we effectively do. This is the reason why the discrepancy between task and activity should be analyzed.

This is where the difficulty lies—systems are no longer only mechanically complicated, they are highly interconnected and, therefore context-dependent and complex. People generally handle variability well, engineered systems do not. Engineered systems are programmed (handling procedures only); people are flexible and creative. More importantly, good engineering designers envision possible future systems that need to be developed; they do this using an abduction process. Engineered systems are excellent at deduction; people are unique at handling induction (Harris, Ballard, Girard, & Gluckman, 1993) and, more importantly, abduction (Peirce & Burks, 1958). Abduction is an inference method that consists in claiming an outcome and figuring out how to reach it (claim B, where you want to be, and demonstrate that A→B, where A is where you are now).

HSI is both a process and a solution. Technological integration, as a process, is often done too late. Discipline-wise, work is usually very well done (e.g., computer scientists and mechanical engineers are very good at their jobs and their performance is accurate, and most of the time perfect). However, interdisciplinary coordination expertise is often lacking. Systems engineering is supposed to ensure such coordination. Why? Because of the lack of time spent on human-centered design from the early stages of design and development.

HCD is now possible because we have very realistic HITLS capabilities, based on virtual prototypes, which enable activity observation and analysis. Virtual prototypes are incrementally developed as digital analogs (i.e., digital twins) of targeted systems (Grieves, 2016; Madni, Madni, & Lucero, 2019). These capabilities drastically change the way system knowledge is acquired. Indeed, systems knowledge was typically acquired when a system was fully developed because testing a fully integrated system is not the same as testing parts of it prior to integration. Consequently, since technological integration can be done from the very beginning, virtually of course, the system can be tested. This is good news. HITLS enables design flexibility before irreversible commitments can be controlled. The challenge, however, remains of testing the tangibility of such solutions.

3 TOWARD A HUMAN-CENTERED SYSTEM SCIENCE

This chapter was perfectly timed to the period when HCD and HSI became possible (Boy, 2013, 2017). This is the reason why, in addition to technological means that support HCD and HSI, it is also a good time to readjust what system science is becoming in a human-centered way (Boy, 2020).

3.1 What Is a System?

In engineering, the term system is used to denote a machine. However, HSI requires that we revisit this common usage. Indeed, medical doctors and physiologists talk about a cardiovascular system or a neuronal system; engineers talk about a propulsion system or a computer system. "System," in one usage, represents a natural entity. Another usage is for an artificial entity. What is the common ground between these two usages of the term system? This section is devoted to answering this question.

The concept of system enables us to represent either a natural or an artificial entity. In AI, where knowledge representation of the world is written in a form of software that can be used for simulation, in HSI, natural and artificial entities are represented in the form of systems defined by their structures and functions, whether physical and/or cognitive, to simulate the natural and artificial entities being so represented. The concept of system as representation paves the way to subjectivity, and also meaning at the same time. Indeed, someone could have a specific idea of what a system is about, and someone else could have a different idea. However, there are common grounds that could be captured and induce generic systemic concepts. The main purpose here is to incrementally build a topology that supports system science.

A natural entity can be a human being, an organ of a human being, a plant, or an animal. An artificial entity can be an abstract object or an abstraction (e.g., a law, a legally defined country, a method), a concrete object (e.g., a chair) or a machine (e.g., a car or a washing machine) that was built by a human being to facilitate the execution of specific tasks.

A system can be either cognitive (or conceptual), physical or both (Boy, 2017). A human being, represented as a system, has cognitive and physical capabilities. A computing system that is capable of perception, inference, and action can also qualify as a cognitive system. A flight management system (FMS), for example, is capable of providing an aircraft with a trajectory based on appropriate data and an air navigation model that enables inferring where to go next correctly. FMS qualifies as a cognitive system. Figure 3 presents a simple ontological definition of the "system" representation.

A system has at least a structure and a function. The human heart has a structure and its physical function is pumping blood. Associated with the human brain, the heart has a regulation cognitive function. Today, machines have cognitive functions (e.g., the cruise control function on a car enables the car to keep a set speed). A computer program or software has a structure that is cognitive. More generally, we can associate hardware with a physical system, as well as software with a cognitive system.

In addition, the conventional single-agent definition of a system function as something that transforms an input into an output (Figure 4) should be extended to a multi-agent perspective.

Artificial intelligence (AI) as a discipline, massively developed during the 1980s, is coming back today. It is interesting to note that the concept of "agent" in AI is very similar to the concept of system as a representation. As already said, an agent, in Minsky's sense, is a society of agents (i.e., an agency is an agent itself). For example, a postman is an agent of an agency that is commonly called "The Post."

Consequently, in this chapter, the term "system" is a synonym of the term "agent." In the same way an agent is a society of agents, a system is a system of systems. Therefore, a system's structures and functions can be defined as structures of structures and functions of functions. More generally, it is now common to use the system-of-systems concept to denote sociotechnical interconnected systems (Figure 5).

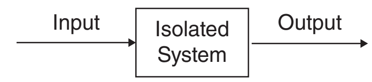


Figure 4 Single-agent isolated system.

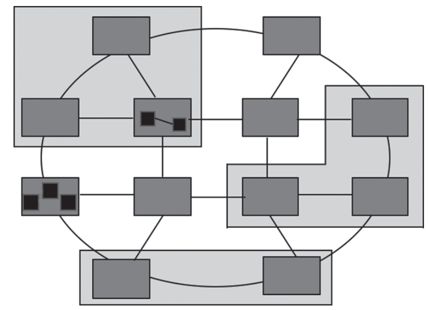


Figure 5 A system is a system of systems (i.e., a system includes an organized set of systems, and a system belongs to a bigger system).

3.2 System's Function Recursive Definition

A typical postman's function can still be defined as having a (prescribed) task of "delivering letters" (i.e., function input), which defines the postman's role in the multi-agent sense (i.e., the postman's role in the postal agency or postal system of systems). Their activity (i.e., function output) may not always reflect such a prescribed task because the environment may change, their capacity may change (e.g., the postman is tired or gets sick) or other contextual factors may change (e.g., heavy rain or excessive traffic jam). This is the reason why a system function should be defined by three attributes:

- a role;
- a context of validity;
- a set of possible and necessary resources.

The context of validity of our postman's role (i.e., delivering letters) can be defined by a time context (e.g., from 8:00 am to noon and from 2:00 pm to 5:00 pm) and a space context (e.g., the

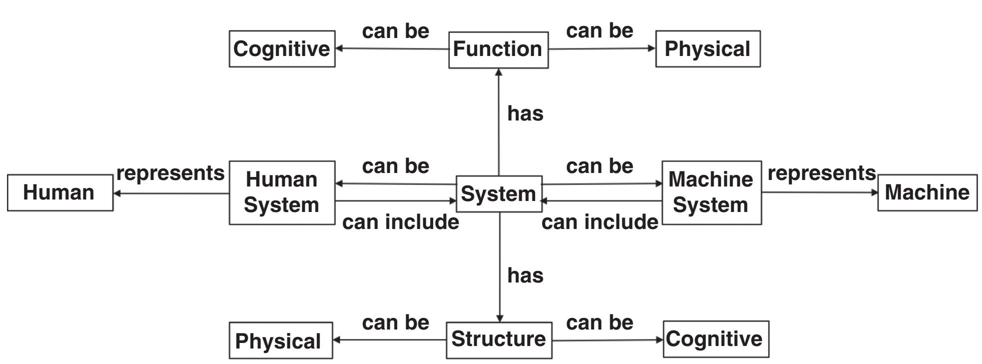


Figure 3 The cognitive-physical structure-function approach of the system representation.

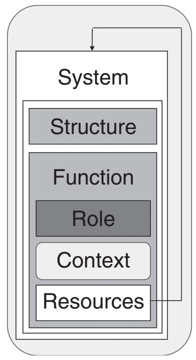


Figure 6 Recursive definition of a system, where resources are systems themselves.

neighborhood where they have to deliver letters). Context can be normal (i.e., every day the same) or abnormal (e.g., some other postmen are absent and they need to expand their time and/or space context).

For example, resources for our postman can be a bag and a bicycle (i.e., physical resources), and a pattern matching cognitive process that enables them to put each letter in the right box (i.e., a cognitive resource). At this point, it becomes clear that a function is a function of functions (e.g., the postman's function to deliver letters is a function of another function, the pattern-matching function). More generically, the function's resources are systems themselves, be they physical, cognitive, or both. Since systems are made of structures and functions, Figure 6 presents a recursive definition of a system. This representation is highly convenient for function allocation in a system of systems (i.e., among systems or structures in a network of systems).

Now, let's assume that the postal services are on strike, which implies that some postmen are not working and therefore the postmen remaining on duty may be required to work longer hours and/or have larger neighborhoods. If the strike is very strict, there might not be enough postmen to assure mail delivery; therefore, the post needs to employ temporary personnel, who have to be trained, supervised and assessed by the existing postmen. Consequently, postmen remaining on duty have to use cognitive functions such as "training," "supervision," and "assessment." A new network of cognitive functions is thus deployed among the new body of postmen.

In the same way, contexts can be represented as contexts of contexts. In aviation, for example, the overall flight context can be decomposed into smaller contexts that include taxiing, takeoff, after-takeoff climb, cruise, descent, approach, landing, and so on. Each of these contexts can be decomposed into even smaller contexts. A system may evolve from Context 1 to Context 2 (Figure 7), where Context 1 may represent a normal situation and Context 2 an abnormal situation, for example.

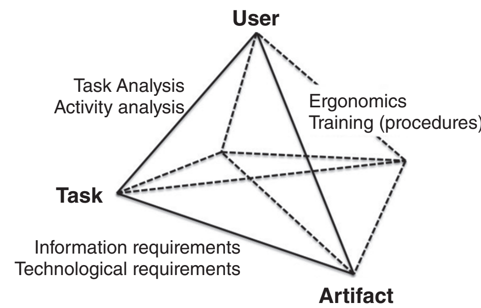


Figure 8 The AUT triangle.

3.3 The AUTOS Pyramid as a Deeper Framework than the TOP Model

The AUTOS pyramid concept was developed as a pragmatic and operational guide for HCD (Boy, 2011):

"A" refers to Artifact (i.e., Technology in the TOP model);
"U" means User (i.e., People in the TOP model);
"T" means Task;
"O" means Organization;
"S" means Situation.

The AUT triangle (Figure 8) associates:

- Artifacts may be cars or consumer electronics systems, devices, and parts, for example.
- Users may be novices, experienced personnel or experts, coming from and evolving in various cultures.[11] They may be tired, stressed, making errors, old or young, as well as in very good shape and mood.
- Tasks may vary from high-level to low-level (e.g., managing a team or an organization, designing, making decisions, and handling quality control). Each task involves one or several cognitive and/or physical functions that related users must learn and use.

The notion of "A-factor" is introduced to represent technology and everything that turns out to be crucial in the use of this technology. A-factors are indicators to U-factors (i.e., human factors). They are commonly expressed in terms of usability, utility, stability, sustainability, and so on.

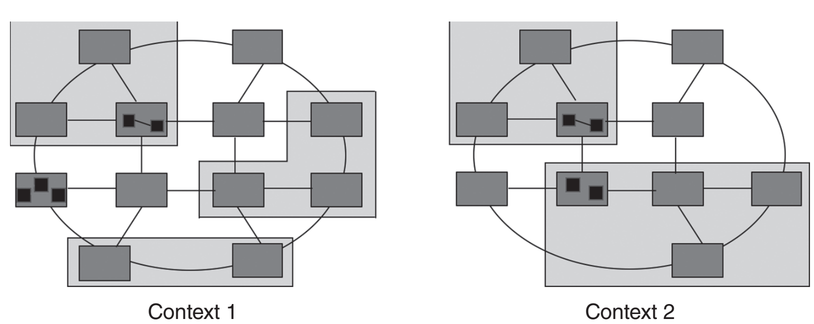


Figure 7 Evolution of system of systems from Context 1 to Context 2.

Let's take the example of an "increasingly autonomous car" (A) that is driven by a "driver" (U). A high-level task (T) might be "driving kids to school in the morning," which can be decomposed into sub-tasks, such as choosing the best route that minimizes time, controlling steering, driving carefully to ensure safety, and so on. This should be tested for a variety of drivers going from novices to expert drivers. Several varieties of drivers should indeed be involved in humans-in-the-loop simulations to discover emergent properties and functions.

It is particularly interesting to test technical documentation complexity because it is directly linked to the explanation of artifact complexity. The harder a system is to use, the more related technical documentation or performance support is required in order to provide appropriate assistance at the right time in the right format.

The AUT triangle enables the explanation of three edges:

- task and activity analysis (U–T);
- information requirements, and technological requirements and limitations (T–A);
- ergonomics and training (procedures) (T–U).

AUT complexity is characterized by content management, information density, and the rules of ergonomics. Content management is, in particular, linked to information relevance, alarm management, and display content management. Information density is linked to decluttering, information modality, diversity, and information-limited attractors, i.e., objects on the instrument or display that are poorly informative for the execution of the task but nevertheless attract the user's attention. The "PC screen do-it-all syndrome" (i.e., the Swiss Army knife metaphor) is a good indicator of the density of both information and functions (elicited improvement-factors were screen size and zooming).

Redundancy is a constant property of a safe, efficient, and comfortable system, be it for repeating information for cross-checking, confirmation, or comfort, or explaining the "how," "where," and "when" an action can or should be performed. Ergonomics rules formalize user friendliness, i.e., consistency, customization, human reliability, affordances, feedback, visibility, and appropriateness of the cognitive functions involved. Human reliability involves human error tolerance (therefore, the need for recovery means) and human error resistance (therefore, the existence of risk to counter). In contrast to these negative sides of people, we also need to investigate the degree of human engagement and involvement. To summarize, A-factors deal with the level of necessary *interface simplicity*, *explanation*, *redundancy*, and *situation awareness* that the artifact is required to offer to users.

Following up on the example of an "increasingly autonomous car" (A) that is driven by a "driver" (U), and having the task of "driving kids at school in the morning" (T), they do this in an organized environment (O), which involves other cars and drivers, passengers in the car, and other stakeholders such as police control ensuring that road rules are correctly followed.

The AUT triangle is limited to the local articulation of the artifact, user, and task. It should be put into perspective within a specific organizational environment, which includes all members of the team that will be represented as systems, and also called "agents." These agents may be humans or machines, interacting with the user who performs the task using the artifact. The introduction of the Organizational environment contributes to our consideration of three additional edges that shape the AUTO tetrahedron (Figure 9) associating:

- social issues (U–O);
- role and job analyses (T–O);
- emergence and evolution (A–O).

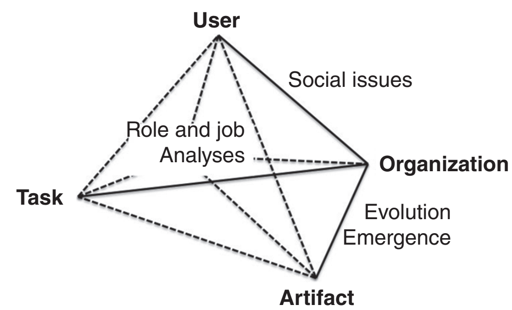


Figure 9 The AUTO tetrahedron.

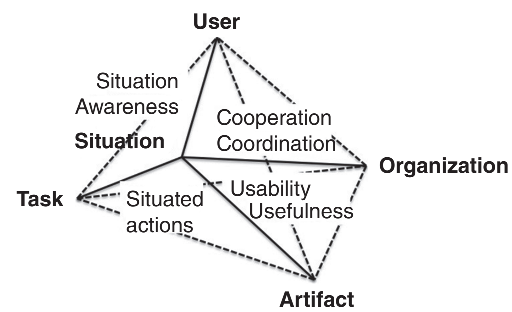


Figure 10 The AUTOS pyramid.

Finally, the AUTOS pyramid framework (Figure 10) is an extension of the AUTO tetrahedron that introduces a new dimension, the "Situation". Its four new edges are:

- usability/usefulness (A–S);
- situation awareness (U–S);
- situated actions (T–S);
- cooperation/coordination (O–S).

At this point, it is useful to clarify the concept of situation. A situation can be viewed in many ways. The following is conceptually based on control theory, where a situation is defined by a set of states. In this chapter, a situation S may refer to a dynamic set of states (i.e., a situation varies in time), $S(t) = \{s_i(t); i = 1,n\}$, including multiple derivatives, in the mathematical sense, such as velocity and acceleration (i.e., a situation is not only a static description, but also an evolution). Let's try to construct a model of these various kinds of situations (Figure 11).

Ideally, the real world is characterized by an infinite number of highly interconnected states. This is what we call the "real situation." It may happen that some of these states are not

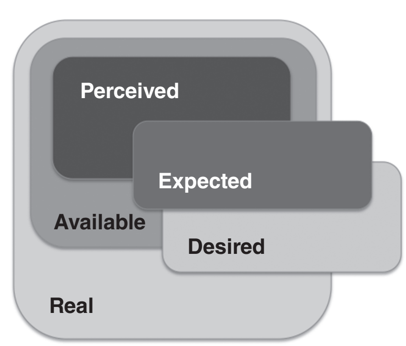


Figure 11 Various kinds of situations.

available to us, either because we cannot access them, or there is no mechanism to make them accessible to us. For example, many states describing aircraft engine health are not directly available to pilots. States available to a human observer define the “available situation” (e.g., aircraft engine health states available to pilots). Note that the “available situation” is typically part of the “real situation.” In addition, the “available situation” may not be perceived in its entirety by the observer. What he/she perceives is called the “perceived situation.” Of course, the “perceived situation” is part of the “available situation,” but is also directed by what is being expected. The “desired” situation typically expresses goal-driven behavior (e.g., what we want to get at). The “expected” situation expresses event-driven behavior (i.e., what we anticipate of a set of states, based on some kind of prediction).

When people expect something to happen with high confidence, they may be confused and mix the “perceived situation” with the “expected situation” (i.e., this is usually related to cultural context, distraction and focus of attention—people see what they want to see!). There is a huge difference between monitoring activities and controlling activities. People involved in a control activity are goal-driven (i.e., they have a strong “desired” situation). Their situation-awareness process is directed by the task they need to perform (i.e., the role they have in the context of where they are). Conversely, people who only have to monitor a process (and who do not have to act on it) need to use, and sometimes construct in real time, an artificial monitoring process that may be difficult, boring, and sometimes meaningless. In this second case, the situation awareness process has great potential not to be accomplished correctly.

Finally, the “perceived situation” is not necessarily a vector of some available states, but a model or image that emerges from a specific combination of these states, incrementally modified over time. This is called experience acquisition. Human operators build their own mental models or mental images of the real situation. This mental image depends on people, cultural context, current activities, and other factors that are specific to the domain under study. You can see here the influence of cognitive context on physical context, because what people perceive is not entirely the real situation but something constructed from the available situation, their own desired situation, as well as background knowledge and skills. For example, in complex situations, such as flying in stormy weather, where time pressure is very high, pilots may multi-task for issues that are difficult to overcome (Loukopoulos, Dismukes, & Barshi, 2001; Wickens & Sebok, 2013).

Consequently, people who are not familiar with complex situations in laboratory setups may produce false interpretations one day or another. For this reason, HCD formative evaluations dealing with complex systems require training, minimal experience acquisition, and longer involvement of human operator subjects. This is behind the recommendation for design teams to develop HCD processes based on real-world experimental setups (e.g., realistic aircraft simulators and professional pilots).

This leads to the definition of two more types of situations: intrinsic and extrinsic. Intrinsic situations are related to the complexity of the human operators’ capabilities. Extrinsic situations are related to the complexity of the human operator’s environment. Both types of situations could be expressed in terms of the number of states and interconnections among these states. In both cases, appropriate models need to be developed. Real and available situations are categorized under the concept of extrinsic situations. Expected and desired situations characterize the concept of intrinsic situation. Perceived situations belong to both concepts of extrinsic and intrinsic situations.

3.4 The Orchestra Model

Organizational analysis requires appropriate models. At this point, the aim of this chapter is not to provide an exhaustive treatment of all models for organizational models, but to present an evolution and a vision for a model that uses the metaphor of the Orchestra.

The metaphorical shift from the Old Army Model (OAM) to the Orchestra Model has been advocated already (Boy, 2013). For a long time, most industrial organizations worked based on the OAM (i.e., a general at the top, then officers, and soldiers at the bottom). OAM information flow is vertical, and mostly top-down. Soldiers do not or barely exchange information horizontally. They are executants of low-level tasks. Information flows are mostly vertical, and mainly descend from the general to the soldiers. Decisions are made at the top. For the last few decades, horizontal information flow exists supported by information technology (e.g., phones, emails, the Internet). This kind of information-technology-based horizontalization happens anywhere, and more specifically in industrial organizations. Consequently, the very concept of OAM has to evolve. In addition, the soldiers of the past are now increasingly specialized. They have become experts in a given field of practice, and even often belong to communities of practice. The overall organization concept is shifting toward the Orchestra Model (Jani & Mehta, 2019).

For an orchestra to be effective, and enable playing a symphony, all musicians are required to have a common frame of reference, which is music theory (e.g., they know how to read and understand the meaning of scores). Who writes scores? This is the dedicated job of composers. A composer of a symphony needs to articulate and coordinate various scores among each other. The composer has to coordinate the tasks of the musicians of the orchestra. More generally, this is task coordination that enables an organization to make some kind of product. At performance time, the orchestra requires a conductor to synchronize the musicians’ activities. In other words, the conductor coordinates a system of systems (i.e., an orchestra of musicians). Musicians themselves are autonomous systems or agents capable of playing their part perfectly, but require coordination with the other musicians. Again, activity (i.e., musical performance) may differ from task (i.e., scores) with respect to various contextual facets (e.g., the sound quality of the room). In addition, musicians also need to cooperate with each other to ensure a reasonable amount of stability, and resilience in case of a musical mistake by one of the musicians: “One for all! All for one!” Another part of the orchestra metaphor is the audience—music stakeholders (i.e., composers, conductors, and musicians) produce pieces of music for potential listeners! More generally, designers, crafters, and engineers generate products for specific people (i.e., we often call them users or human operators). The audience brings issues such as acceptability, usability, and usefulness. The audience includes anybody (i.e., a general public audience) or experts.

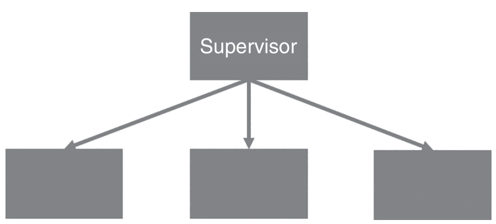
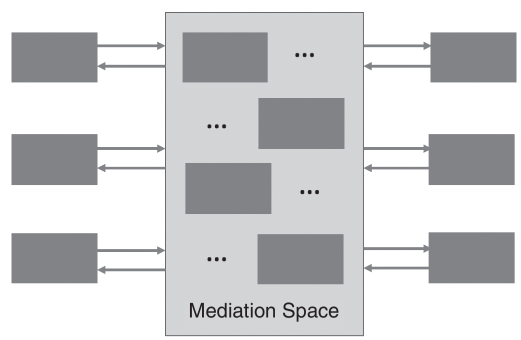
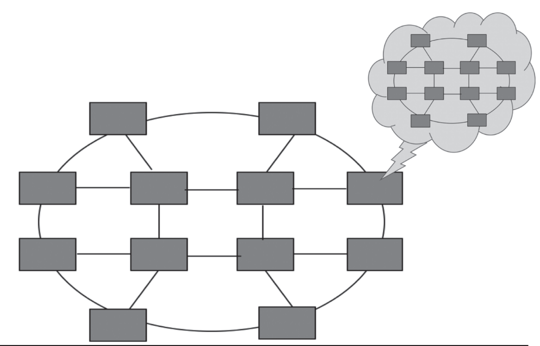
In the same way that we have several types of orchestras (e.g., symphonic, jazz), current industrial organizations may take several forms (e.g., highly structured and large, loosely structured and small). Structured and large organizations are typically based on proceduralized functions (e.g., symphonic orchestra). Loosely structured and small organizations are typically based on problem-solving functions (e.g., a jazz band). In all cases, we are facing agencies of agents (i.e., systems of systems (SoS) or teams of teams), and the more agents are autonomous, the more the agency should be coordinated.

In addition to the Orchestra Model, it is necessary to define finer-grained systemic interaction models that are based on various levels of intersubjectivity (Frie & Reis, 2001), among agents in a society or agency (i.e., what they can share and mutually understand).

3.5 Systemic Interaction Models

There are three main kinds of systemic interaction models (Table 1): supervision; mediation; and cooperation.

Table 1 Systemic Interaction Models

Supervision of systems by a system	Mediation among systems through a mediation space	Systems cooperating among each other thanks to their knowledge of the other systems
		

Supervision is when a system (i.e., a supervisor) oversees interactions among other systems. Supervision is about coordination. This interaction model is used when systems do not know each other or do not have enough resources to properly interact with each other toward a satisfactory performance of the systems of systems (SoS) that they constitute.

Mediation is when systems are able to interact with each other through a mediation space composed of a set of mediating systems (i.e., like diplomats). This interaction model is used when systems barely know each other, but easily understand how to use the mediation space.

Cooperation is when systems are able to have a socio-cognitive model of the SoS which they are part of. Each system uses its socio-cognitive model to interact with the other systems to maximize some kinds of performance metrics. Note that this principle can be collective and democratic. Other principles could be used such as dominance of a system over the other systems (i.e., a dictatorial principle). The cooperation interaction model is used when systems know each other through their own socio-cognitive model, which is able to adapt through learning from positive and negative interactions. Up to now, cooperation has been mostly a capability of humans. However, artificial intelligence (AI) brings new ways of providing machines with such cooperation capabilities. AI can provide situation awareness, decision-making and planning capabilities and support in specific contexts. In addition, AI can also provide machines with learning capabilities, and more specifically the possibility of upgrading system's socio-cognitive model from experience, in specific contexts. Note also that cooperation requires coordination, and situation awareness is key (Endsley, 2019; Stanton, Walker, Salmon, & Hancock, 2017).

Using these three systemic interaction models, it is clear that systems, as agents, become more autonomous when they go from being supervised to being mediated to cooperating with each other.

4 PUTTING HUMAN SYSTEMS INTEGRATION INTO PRACTICE

In the title of this chapter, integration comes before design. The reason for this was to keep "human systems integration" together as both an approach and a generic solution for human-centered engineering design. In this section, HSI and design (HSI&D) will be described practically, based on theoretical knowledge presented in Section 3. Six crucial concepts will be emphasized: domain experience; modeling; complexity; HITLS; performance metrics; and experimental protocols.

4.1 Experience-Based Participatory Design

Designing new life-critical technology is not only a matter of creativity, but also a matter of domain experience. Creativity and experience are often two contradictory concepts; the former being a matter of integration, and the latter based on practice. For a graphic artist, creating a new color, such a new orange, requires combining other existing colors, such as red and yellow. The artist starts by combining available tones of red and yellow colors, and continues to iterate until a satisfactory shade of orange is found. Of course, success in such a process strongly hinges not only on experience in the choice of existing colors, but also on the art of incrementally assessing results. This is the reason why artists practice regularly to become proficient in their creative art.

Innovation requires creativity and experience. We cannot create a new aircraft engine without experience in the aeronautical domain, and more specifically in propulsion. However, such experience may become counterproductive if it is used without considering possible designs that were never thought of at this point. Some possible future systems can be imagined and tested via HITLS to explore potential usages (i.e., activities). Design and development should be considered as abduction processes (i.e., processes in which one projects into the future and tests the validity and relevance of a socio-technical solution). Tests should be performed by or with subject matter experts (i.e., people who are recognized as having experience in a domain).

4.2 Model-Based Methodology

Modeling is at the heart of HSI&D. Nicolas Boileau, a French poet and leading literary critic of the seventeenth century, said. "Whatever is well conceived is clearly said, and the words to say it flow with ease" (Boileau, 1674, reprinted in 1966). Modeling consists in this kind of exercise that consists in representing existing or potentially credible information on and/or knowledge about a given domain. For example, astronomers tried to model planet trajectories for centuries (i.e., represent in a concise and appropriate manner the way planets evolve in space).

Newton tried to model the motion of a planet (revolving around the Sun) by defining the force acting on this planet as directly proportional to its mass and inversely proportional to the square of its distance from the Sun. He discovered the theory of universal gravitation. How did he find it? He first tried to answer the question: "What force causes the planets to revolve around the sun?" He also tried to answer the question, "Why are their orbits elliptical?" These questions were based on previous claims by Copernicus, Tycho Brahe, and Kepler, one or two centuries before. Again, this association between creativity and experience must be followed by abductive demonstration (i.e., claiming B, demonstrate that $A \rightarrow B$; e.g., claiming "orbits are elliptical," and demonstrate that "universal gravitation theory $\rightarrow$ orbits are elliptical").

Like Newton, Copernicus, Tycho Brahe, and Kepler, we would like to have HSI&D claims involving models that need to be validated through consistency checks and reference to observed datasets. Life-critical systems are usually analyzed, designed, and assessed based on metrics related to safety, efficiency, and comfort. Of course, this list of metric types is not exhaustive, and requires extension. Problems come from the choice and effectivity of the right human and machine factors, as well as potential correlations between these factors and safety, efficiency, and comfort. In human factors and ergonomics, workload, situation awareness, and decision making are typically tested. In human–computer interaction, usability and usefulness are main factors to be tested. In HSI, factors producing systemic complexity are important to consider.

4.3 Human Systems Complexity Analysis

Complex systems are usually determined by the following properties (Boy, 2017):

1. a large number of components and interconnections among these components;
2. many people are involved in their life cycles including design, development, manufacturing, operations, maintenance, and decommissioning;
3. emergent properties and behaviors are not included in the components;
4. complex adaptive mechanisms and behaviors: this can be called adaptability;
5. nonlinearities and possible chaos: this can be called unpredictability. In this chapter, complex systems include people and machines. They are "complex sociotechnical systems" (Baxter & Sommerville, 2011; Carayon, 2006; Grudin, 1994; Norman & Stappers, 2016).

Examples of complex systems are aircraft, industrial power plants, and large defense systems, which typically involve multiple expertise to design, manufacture, use, repair, and dismantle them. In contrast, simple systems can be defined by the following properties:

1. small number of components and interconnections;
2. behavior directly related to components;
3. no or very simple adaptive mechanisms and behavior;
4. linear or slightly linear responses to inputs. Examples of simple systems are tables, cars, and electronic watches. They do not require the involvement of many people, except in the case of mass production.

From a design perspective, we will distinguish and combine structural and functional complexity. The former is related to system structure, sub-structures, and so on. The latter is related to system function, sub-functions, and so on. For example, the life cycle of an aircraft involves complex processes that deal with a large number of complex systems. Therefore, several sub-systems need to be articulated and coordinated structurally and functionally. Consequently, several articulated backgrounds are required to design and manufacture a complex system, such as an aircraft. There is no space for improvisation. Whether they are designers, manufacturers, or human operators, people who deal with complex systems need appropriate levels of familiarity with them and the environment they induce. This is what human-systems complexity analysis is about.

Finally, the HCD of complex systems is necessarily interdisciplinary, since no-one can provide all possible contributions to the design of such systems, but a well-formed team can. Therefore, collaborative work is an important part of HCD (Grudin & Poltrock, 2003). These concepts will be further developed in the chapter, using my concrete aerospace experience to illustrate them and make them more tangible.

4.4 Human-in-the-Loop Simulation

HITLS has become a common practice in human-centered design (Boy, 2013, 2020; Rothrock & Narayanan, 2011). HITLS is useful to enable activity observation and therefore analysis. Let's illustrate HITLS for the construction of a house. If you decide to build a house, you would want an architect. This specialist will interview you to get to know you better, to understand your needs, sense your preferences, and so on. They will design and develop a mockup of your dream house, using what is elicited from you. They will show you the mockup and observe your reactions, or even better your activity in the virtual house (e.g., using a virtual headset). They will analyze what you like and dislike in it, and will iterate the right house for you.

HITLS cannot be set up before appropriate scenarios are defined and correctly developed. This is the reason why knowledge elicitation methods should be used to acquire procedural and declarative scenarios (i.e., operational knowledge) from subject matter experts (Martinie, Palanque, Ragosta, & Fahssi, 2013). Operational knowledge is typically expressed in the form of timelines (i.e., scripts as in theater plays) and configurations (i.e., theater scene definition in terms of actual actors and objects, or actual human and machine agents). This operational knowledge also helps in defining system performance. With increasingly autonomous machines, we talk about "human machine teaming" (UK MOD Chiefs of Staff, 2018). Indeed, such machines have behaviors that need to be identified and included in experimental scenarios.

4.5 System Performance and Associated Metrics

System performance relies on the articulation of three factors encapsulated within the TOP Model (Boy, 2011, 2013): (1) technological (e.g., usability, explainability, etc.); (2) organizational for teamworking (e.g., trust and collaboration, etc.); and (3) human (e.g., workload, stress, memory, etc.). HSI can be investigated by combining interdisciplinary knowledge and knowhow for technology (e.g., aeronautics, artificial intelligence and human–computer interaction), human and organizational factors, field expertise, that will enable the assessment of human–machine teaming performance.

Let's take the MOHICAN project as an example of the establishment of a performance model that supports the development and use of performance metrics (Boy, Dezemery, Haffreingue, Lu Cong Sang, & Morel, 2020). MOHICAN's objectives are threefold.

1. A multi-agent model supports information processing and transfer among the various human and machine systems (or agents) based on tactical scenarios and contexts, as well as required physical and cognitive functions, {Fc} = {Role, Context of validity, multi-agent Resources}. This model will be used both to guide performance monitoring and discover emerging cognitive functions.
2. A three-layer assessment model, based on iterative human-in-the-loop experimental simulations (essential for Agile design and development of human system integration studies), that will integrate measures: (a) low-level measures $\{m_i\}$ that can be objective (e.g., eye tracking data, military performance data) and/or subjective (e.g., Cooper-Harper evaluation scales, NASA TLX) measures, as well as a posteriori analysis of agents' activities (e.g., self-assessment of recorded flight scenes), into (b) human factors criteria $\{C_j\}$ (e.g., workload, fatigue, attention, vigilance, engagement, affordances, flexibility, maturity [technological, organizational and human], tangibility). The model $C_j = g(\{m_i\})$ is developed using the cognitive function analysis method (Boy, 1998) extended by operational performance criteria (e.g., risk management, task achievement, operational margin). (c) Teamwork performance will then be modeled using shared situation awareness and human–machine cooperation through teamwork metrics, $\{T_k\}$ [$T_k = {}_{fk,context}(\{C_j\})$] within contextualized use cases. Examples of such metrics are trust and collaboration.
3. A method of T_k qualification to check the consistency and pertinence of team performance with the measured military operational performance.

5 SOCIOTECHNICAL IMPLICATIONS AND PERSPECTIVES

HSI&D can only be used once an organization has adapted itself to it. This is the reason why organizational changes should be developed (i.e., the organization should be redesigned in a human-centered way). When we talk about "organization," we talk about both the organization that is designed and developed, and the organization that will be the solution (i.e., the targeted new socio-technical system).

5.1 Reshaping Organizational Setups and Operations Practices

In a multi-agent world, it is crucial to master the way systemic interaction models are correctly allocated, not only initially, but also dynamically. Without HITLS, this is an impossible HSI&D goal. Both systemic physical and cognitive redundancy are required at all possible stages of a system of systems (SoS).

Such redundancy should be organized to provide immunological properties (e.g., Built-In Test Equipment on aircraft) and homeostatic defenses to the SoS (e.g., local adaptation processes).

Regarding integration for safety, efficiency, and comfort, near-misses acquisition and analysis are mandatory during the whole life cycle of life-critical complex systems. However, major accidents do not arise due to a lack of incidents, they come from complacency, over-trust, and lack of collaboration. Complacency typically results from over-confidence based on past successes, focusing instead on systematic verification practices.

This is the reason why we constantly need to maintain awareness using appropriate high-potential indicators providing levels of seriousness for a variety of situations (see above for the various categories of situations). Life-critical complex systems need to be thought out and handled with crisis management in mind any time it is required (i.e., always performing the following processes: prevention, recovery, and mitigation of consequences), in addition to following procedure. This requires life-critical-situation awareness (LCSA), and shared LCSA in multi-agent systems. LCSA brings a crucial factor related to the possibility of death. Hutchins's distributed cognition anthropological approach is very well suited here (Hutchins, 1995a). Hutchins studied pilots as a social anthropologist, finding behavioral and cultural patterns and meaning. Again, intersubjectivity cannot be properly studied and understood without a distributed cognition approach and a multi-agent representation.

Without experience feedback, all this is impossible. Life-critical situations emerge from activity. This is the reason why we need to cultivate and use an experience feedback spirit. Experience feedback should be present and active at all levels of SoS. Collaboration is key. However, early human-systems integration allows for a truly optimized future system when designing complex sociotechnical systems (Boeing, Cham, Jorritsma, & Griffin, 2019). This is the reason why HITLS are so important in HSI.

5.2 Designing for Systemic Flexibility

For a long time, machines were automated based on technological opportunities (i.e., we automate because it is possible). High technology, and more specifically information technology, developed tremendously in the last three decades of the twentieth century. As already mentioned above, automation can be described as the transfer of cognitive functions from people to machines (e.g., pilots were manually in charge of aircraft handling qualities; a task that recently has been transferred to computers). However, such an automation of machines was previously handled using procedures, which "automated people" (Figure 12). Both kinds of automation are now mature and have greatly improved safety and efficiency in aviation, for example, but they are working within specific contexts, which induce operational rigidity. Outside of these contexts, when an unexpected situation occurs or the current context does not fit the automation context of validity (Pinet, 2015), pilots have to solve problems by themselves, and therefore require more autonomy and flexibility. This evolution from rigid automation to flexible autonomy is illustrated in Figure 12, where autonomy involves multi-agent problem solving (i.e., cooperation among human and machine agents) and therefore coordination among these agents.

It becomes clear that following procedure, use of automation, and problem solving are the three main functions useful

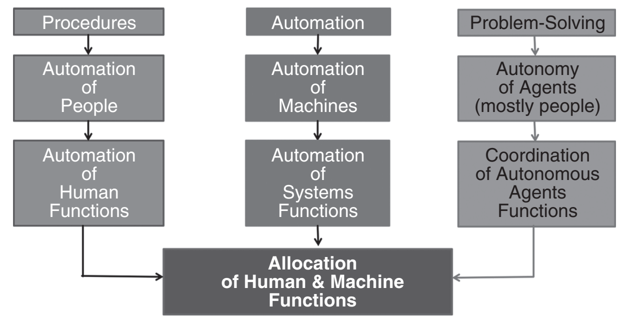


Figure 12 From rigid automation to flexible autonomy.

in the management of life-critical systems. If we have done a lot to support the first two, a wide range of efforts have to be made to support the third one. A question is: what are the situations where people need to solve problems that have not already been compiled into appropriate procedures and automata? We often talk about unexpected situations (Boy, 2013; Pinet, 2015). In such situations, people in charge should have autonomous capabilities. Again, collaboration among knowledgeable agents is a very precious resource. For that matter, these agents, be they humans or machines, need to be appropriately coordinated. Altogether, designing for systemic flexibility is a matter of good human and machine function allocation.

5.3 Virtual HCD Provides More Flexibility and System Knowledge

It is clear that we now design and develop based on massive amounts of software and data. Therefore, modeling and simulation, more specifically HITLS, are at the center of design processes. This is good news since activity can be tested instantly at the design stage, and consequently human factors and ergonomics can be seriously considered in an agile way in systems being designed and developed. However, the tangibility issue also needs to be seriously considered (Boy, 2016). Tangibility taken at the individual level can be physical and/or cognitive (figurative), but what are its socio-technical implications? The tangibilization approach presented in Figure 2 should be considered.

Considering a technology-centered system-engineering approach, where everything is physical from the start, resource commitments are also set from the start, which severely constrain design options (Figure 13). Design flexibility becomes a problem very quickly and we do not have enough resources for compensation. We learn about the system when we can use it in the real world, that is when it is almost finished. The only resource that we have is the famous "user interface." However, if the user interface could be designed to make the life of users easier, then it is often developed to compensate design flaws, from an HCD point of view, of course!

Instead, if we use HITLS from the beginning, that is designing and developing virtual prototypes of the targeted system, we can test and analyze an activity, and therefore learn about the system during design and development (Figure 14). When a stakeholder asks, "What will it look like?" You are able to show the simulation and get early system knowledge at the same time since HITLS enables interaction with the simulated system. You can look at what needs to be changed. You still have resource options. You retain design flexibility.

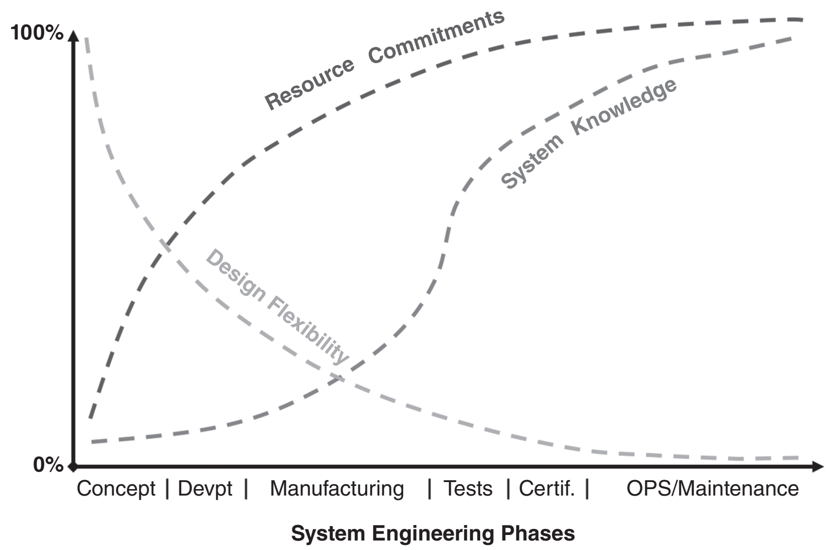


Figure 13 Resource commitments, design flexibility, and system knowledge resulting from a technology-centered system-engineering approach. (*Source:* Adapted from Mike Conroy, 2016.)

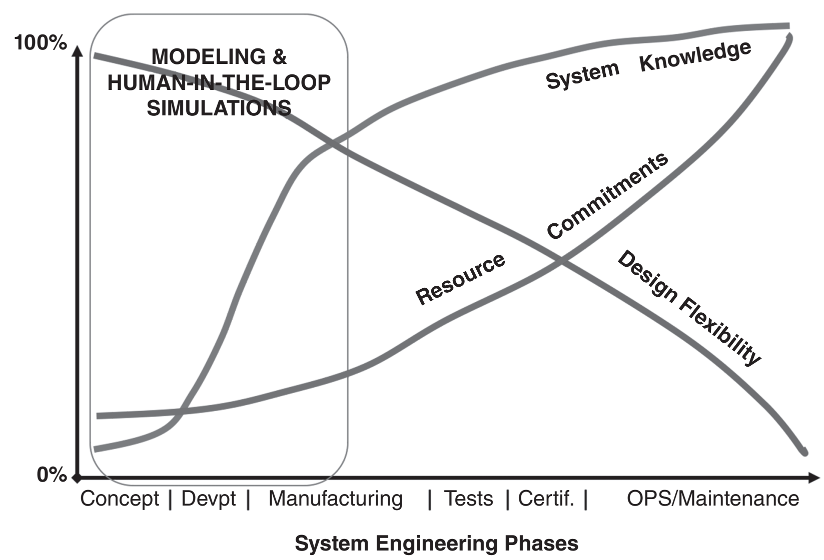


Figure 14 Resource commitments, design flexibility, and system knowledge resulting from a human-centered design approach.

The most important feature in the evolutions of all three parameters (i.e., system knowledge, design flexibility, and resource commitments) versus the life cycle of a system (i.e., systems engineering phases) is the consideration of all three at the same time (Figure 14).

5.4 Human Systems Maturity

Achieving a good HSI using HCD and HITLS means that such Agile development process enables delivery of a tangible system at a given stage of maturity. What do we mean by maturity? Usually, we talk about technology readiness levels (TRL), but these levels are technically based on design and development process quality. In HSI, we introduce three interconnected concepts to assess maturity:

- technology maturity (i.e., related technology robustness, stability, controllability, and observability);
- maturity of practice (i.e., related to people intentionality and reactivity);
- societal maturity (i.e., related to culture and organizations).

Testing maturity means discovering and rationalizing emergent behaviors, functions, and sometimes structures. These emergent properties can be technological, human-related (or job-related), and societal (or cultural).

Agile development is scenario-based (i.e., stories feed into virtual prototypes and test cases). In other words, two orthogonal types of scenarios are incrementally defined and further tested using virtual prototypes and test cases:

- declarative scenarios that represent structural configurations (or infrastructures);
- procedural scenarios that represent functional chronologies (or stories).

The choice of such scenarios can, of course, be very difficult. At this point, scenario definition is more an art than a technique, strongly based on domain experience and expertise (Pinet, 2019). Indeed, storytelling draws on expertise and experience. It is always difficult and often impossible to quantify such expertise and experience, but it is extremely useful to qualify them to guide design decision-making processes.

6 CONCLUSION

This chapter presented HSI&D, which is a blend of Human-Centered Design (HCD) and Systems Engineering (SE) toward Human Systems Integration (HSI). As already seen, HSI&D is not only a matter of technology, it is also a matter of organization and people (i.e., the TOP model). Integration should be considered from the very beginning of the life cycle of a system, and it is never finished until dismantling. Attention should be brought to the way people and technology interact, complement each other, and are able to achieve goals anticipated by system's purpose. In addition, HSI&D should address the sustainability of systems being developed as a major issue (Vega-Mejia, Montoya-Torres, & Islam, 2016).

In sum, HSI denotes the processes and results of contemporary HCD that consist in concurrently considering technological, organizational, and human systems at design stage and operations stage. After a reminder of the evolution of human-centered approaches, this chapter emphasized the overall question of systems from a human-centered system science point of view. The concept of "system" is considered a representation of people and machines cognitively and physically defined in terms of structures and functions. Systems' physical and cognitive attributes can be defined to satisfy a set of principles and criteria, leading to a set of requirements in terms of tasks and activities.

Finally, this chapter proposes a foundation for HSI, articulated between HCD and SE. It should be a departure toward more fundamental development, as well as hands-on practice using currently developed concepts, methods, and tools. Human Factors and Ergonomics (HFE) specialists have already contributed to the development of HSI (Boehm-Davis, Durso, & Lee, 2015), but we need to expand HSI&D to systems engineering and computer science disciplines (Figure 15). The following perspectives are then recommended:

- Use modeling and human-in-the-loop simulations to improve system knowledge very early during the design and development process supported by the TOP model and the AUTOS pyramid, and consequently improve the flexibility of design (and redesign), as well as resource management.
- Put together foundations of a systemic topology/ontology-based approach for HSI, which should be supported by a theoretical and industrial joint approach, via the development and capitalization of real-world tangible cases.

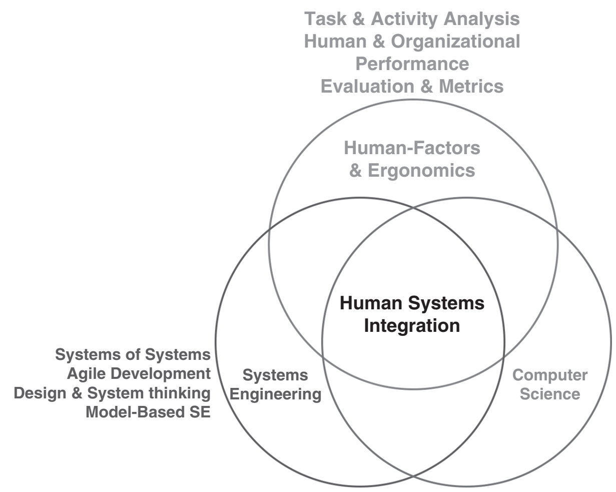


Figure 15 Human systems integration as an intersection of human factors and ergonomics, systems engineering, and computer science.

- Educate and train engineering and human and social sciences students in HSI (e.g., HCD and cognitive engineering, complexity analysis, organization design and management, modeling and simulation, life-critical systems, and advanced interaction media).

NOTES

1 Next Generation Air Transportation System.
2 French acronym for Model of Crew and Aircraft Sub-Systems for Equipment Management.
3 Association for Computing Machinery, Computer Human Interaction conference.
4 In 1982, CHI was called the Human Factors in Computer Systems Conference.
5 See http://2020.hci.international
6 Science, Technology, Engineering and Mathematics.
7 Science, Technology, Engineering, Arts and Mathematics.
8 See http://www.jnd.org/dn.mss/logic_versus_usage_the_case_for_activity-centered_design.html
9 International Council on Systems Engineering.
10 Men Are Better At - Machines Are Better At.
11 Even if I prefer to use the term "people," the tradition in human-machine systems is to use "human operators," especially in the process control and human engineering community, or "users," in the HCI community.

REFERENCES

Agile Manifesto (2015). http://www.agilemanifesto.org. (retrieved on January 26, 2020).

Baxter, G. D., & Sommerville, I. (2011). Socio-technical systems: from design methods to systems engineering. *Interacting with Computers, 23*(1), 4-17. DOI: 10.1016/j.intcom.2010.07.003.

Billings, C. E. (1996). *Aviation automation: The search for a human-centered approach*. Mahwah, NJ: Erlbaum.

Bødker, S. (1996). Creating conditions for participation: Conflicts and resources in systems design. *Human Computer Interaction, 11*(3), 215–236.

Boehm-Davis, D. A., Durso, F. T., & Lee, J. D. (Eds.) (2015). *Handbook of human systems integration*. Washington, DC: American Psychology Association.

Boeing, A. A., Cham, B. S., Jorritsma, K., & Griffin, M. A. (2019). A socio-technical systems approach to the design of future complex systems: A case study of future submarines. Paper presented at INCOSE 2019 International Conference on Human Systems Integration, Biarritz, France.

Boileau, N. (1674, republished in 1966). The art of poetry (in French, *L'Art Poétique*), Canto I, l. 153. In N. Boileau (1966). *Œuvres complètes,* éd. Fr. Escal, Paris: Gallimard.

Borah, J., Young, L. R., & Curry, R. E. (1988). Optimal estimator model for human spatial orientation. In V. Henn & B. Cohen (Eds.), *Representation of three-dimensional space in the vestibular, oculomotor, and visual systems*. New York: The New York Academy of Sciences.

Boulnois, S. L. P. (2018). Human-centered design of a 3D-augmented strategic weather management system for the aviation community. Doctoral dissertation, Florida Institute of Technology, USA.

Boy, G. A. (1983). Le Système MESSAGE: un premier pas vers l'analyse assistée par ordinateur des interactions homme-machine [The MESSAGE system: A first step toward computer-supported analysis of human-machine interactions]. *Le Travail Humain, 46*(2).

Boy, G. A. (1998). *Cognitive function analysis*. Westport, CT: Praeger.

Boy, G. A. (Ed.). (2011). *Handbook of human-machine interaction: A human-centered design approach*. Boca Raton, FL: CRC Press.

Boy, G. A. (2013). *Orchestrating human-centered design*. London: Springer.

Boy, G. A. (2016). *Tangible interactive systems*. London: Springer.

Boy, G. A. (2017). Human-centered design of complex systems: An experience-based approach. *Design Science Journal,* 3.

Boy, G. A. (2019). FlexTech: From rigid to flexible human-systems integration. In R. S. Kennett, R. S. Swarz, & A. Zonnenshain (Eds.), *Systems engineering in the fourth industrial revolution: Big data, novel technologies, and modern systems engineering*. Hoboken, NJ: Wiley.

Boy, G. A. (2020). *Human systems integration: From virtual to tangible*. Boca Raton, FL: CRC Press.

Boy, G. A., Dezemery, J., Haffreingue, B., Lu Cong Sang, R., & Morel, C. (2020). MOHICAN: human-machine performance monitoring through trust and collaboration analysis. Towards smarter design of a virtual assistant and real time optimization of machine behavior. In *Proceedings of ICCAS 2020, the first International Conference on Cognitive Aircraft Systems, ISAE-SUPAERO,* Toulouse, France, March 18-19.

Boy, G. A., & McGovern Narkevicius, J. (2013). Unifying human centered design and systems engineering for human systems integration. In *Proceedings of CSDM'13*, Paris. France.

Boy, G. A., & Tessier, C. (1985). Cockpit analysis and assessment by the MESSAGE Methodology. In *Proceedings of the 2nd IFAC/IFIP/IFORS/IEA Conference on Analysis, Design and Evaluation of Man-Machine Systems, Villa-Ponti, Italy, September 10–12* (pp. 73–79). Oxford: Pergamon Press.

Buxton, B. (2010). *Sketching user experiences: Getting the design right and the right design*. Burlington, MA: Morgan Kaufmann.

Brown, T. (2008). Design thinking. *Harvard Business Review, 86*(6), 84–92.

Carayon, P. (2006). Human factors of complex sociotechnical systems. *Applied Ergonomics, 37*(4), 525–535. DOI: 10.1016/j.apergo.2006.04.011.

Card, S. K., Moran, T. P., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Conroy, M. (2016). Playing nice across time & space: tools, methods and tech for multi-location multi-decadal teams. Paper presented at INCOSE International Workshop, HSI Working Group Meeting, Torrence, California, USA.

Corker, K. (1994). Man-machine integration design and analysis system (MIDAS) applied to a computer-based procedure-aiding system. In *Proceedings of the 38th Annual Meeting of the Human Factors and Ergonomics Society, USA*.

Edwards, E. C., & Kasik, D. J. (1974). User experience with the CYBER graphics terminal. In *Proceedings of VIM-21, October (pp. 284–286)*.

Endsley, M. R. (2019). Human teaming with autonomous systems: Building support for situation awareness. Keynote paper presented at INCOSE International Conference on Human Systems Integration (HSI2019), Biarritz, France.

Fitts, P. M. (Ed.). (1951). *Human engineering for an effective air-navigation and traffic-control system*. Washington, DC: National Research Council.

Flores, F., Graves, M., Hartfield, B., & Winograd, T. (1988). Computer systems and the design of organizational interaction. *ACM Transactions on Office Information Systems, 6*, 153–172.

Frie, R., & Reis, B. (2001). Understanding intersubjectivity: Psychoanalytic formulations and their philosophical underpinnings. *Contemporary Psychoanalysis Journal, 37*(2), 297–327. https://doi.org/10.1080/00107530.2001.10747081.

Garcia Belmonte, N. (2016). *Engineering intelligence through data visualization at Uber*. https://eng.uber.com/data-viz-intel/. Retrieved on July 29, 2019.

Gore, B. F., Hooey, B. L., Wickens, C. D., Sebok, A., Hutchins, S., & Salud, E. (2009). Identification of NextGen air traffic control and pilot performance parameters for human performance model development in the transitional airspace. NASA Final Report: NRA #NNX08AE87A. San Jose, CA: San Jose State University.

Gore, B. F., & Jarvis, P. (2005). New integrated modeling capabilities: MIDAS's recent behavioral enhancements. In *SAE Transactions - Journal of Passenger Cars: Electronic and Electrical Systems (pp. 752–760)*.

Gore, B. F., Verma, S., Jadhav, A., Delnegro, R., &. Corker, K. M. (2002). *Human error modeling predictions: Air MIDAS human performance modeling of T-NASA*. San Jose, CA: San Jose State University.

Grieves, M. (2016). Origins of the digital twin. Concept Working Paper. August. Florida Institute of Technology/NASA, DOI:10.13140/RG.2.2.26367.61609.

Grudin, J. (1993). Obstacles to participatory design in large product development organizations. In A. Namioka & D. Schuler (Eds.), *Participatory design. Principles and practices* (pp. 99–122). Hillsdale, NJ: Lawrence Erlbaum Associates.

Grudin, J. (1994). Computer-supported cooperative work: History and focus. *Computer, 27*(5), 19–26.

Grudin, J., & Poltrock, S. (2003). Collaboration technology in teams, organizations, and communities. Paper presented at Conference: Human-Computer Interaction INTERACT '03: IFIP TC13 International Conference on Human-Computer Interaction, Zurich, Switzerland.

Harris, S. D., Ballard, L., Girard, R. & Gluckman, J. (1993). Sensor fusion and situation assessment: Future F/A-18. In A. H. Levis & I. S. Levis (Eds.), *Science of command and control: Part III Coping with change*. Fairfax, VA: AFCEA International Press.

Hollnagel, E. (1993). Human reliability analysis: Context and control. In B. R. Gaines & A. Monk (Eds.), *Computers and people*. New York: Academic Press.

Hollnagel, E., & Woods, D. D. (2005). *Joint cognitive systems: Foundations of cognitive systems engineering*. New York: Taylor & Francis.

Hutchins, E. (1995a). *Cognition in the wild*. Cambridge, MA: MIT Press.

Hutchins E. (1995b). How a cockpit remembers its speeds. *Cognitive Science, 19*, 265–288.

Jani, G., & Mehta, T. (2019). Organizational process reengineering: A human-centered design approach. Paper presented at INCOSE 2019 International Conference on Human Systems Integration, Biarritz, France.

Kaptelinin, V., & Nardi, B.A. (2006). *Acting with technology: Activity theory and interaction Design*. Cambridge, MA: MIT Press.

Kelley, D. (2016). From design to design thinking at Stanford and IDEO. *Journal of Design Economics and Innovation*, *2*(1), 88–101. DOI: 10.1016/j.sheji.2016.01.009.

Kim, S. Y., Wagner, D., & Jimenez, A. (2019). Challenges in applying model-based systems engineering: human-centered design perspective. Paper presented at INCOSE 2019 International Conference on Human Systems Integration, Biarritz, France.

Kruchten, N. (2018). Data visualization for artificial intelligence, and vice versa.: https://medium.com/@plotlygraphs/data-visualization-for-artificial-intelligence-and-vice-versa-a38869065d88. Retrieved on July 30, 2019.

Letondal, C., Vinot, J. L., Pauchet, S., Boussiron, C., Rey, S., Becquet, V., & Lavenir, C. (2018). Being in the sky: framing tangible and embodied interaction for future airliner cockpits. In *Proceedings of TEI 2018 12th International Conference on Tangible, Embedded and Embodied Interactions, March, Stockholm, Sweden.* 10.1145/3173225.3173229. hal-01715317.

Loukopoulos, L. D., Dismukes, R. K., & Barshi, I. (2001). Cockpit interruptions and distractions: A line observatory study. In *Proceedings of the 11th International Symposium on Aviation Psychology*, March, Columbus, OH.

Madni, A. M., Madni, C. C. & Lucero, S. D. (2019). Leveraging digital twin technology in model-based systems engineering. *Systems.* doi:10.3390/systems7010007.

Martinie, C., Palanque, P., Ragosta, M., & Fahssi, R. (2013). Extending procedural task models by systematic explicit integration of objects, knowledge and information. In *Proceedings of the 31st European Conference on Cognitive Ergonomics (pp. 1–10).*

McRuer, D. T., & Krendel, E. S. (1974). *Mathematical models of human pilot behavior. AGARDograph No. 188, NATO*.

Minsky, M. (1986) *The society of mind*. New York: Simon & Schuster.

Moertl, P., Neuhuber, N., & Pretto, P. (2019). A human-systems integration framework around the SAE taxonomy for automated driving systems. Paper presented at INCOSE 2019 International Conference on Human Systems Integration, Biarritz, France.

Montebelli, A., Herrera, C., & Ziemke, T. (2007). An analysis of behavioral attractor dynamics. *Advances in artificial life, Lecture Notes in Computer Science, vol. 4648, (pp. 213–222). Berlin: Springer Verlag*: http://dx.doi.org/10.1007/978-3-540-74913-4_22.

Muller, M. J. (2007). Participatory design: The third space in HCI (revised). In J. Jacko & A. Sears (Eds.), *Handbook of HCI* (2nd ed.). Mahwah, NJ: Erlbaum.

Nielsen, J. (1993). *Usability engineering*. Burlington, MA: Morgan Kaufman.

Norman, D. A. (1986). Cognitive engineering. In D.A. Norman & S. W. Draper (Eds.), *User centered system design*. Hillsdale, NJ: Erlbaum, pp. 31–61.

Norman, D. (1988). *The design of everyday things*. New York: Basic Books.

Norman, D. (2013). *The design of everyday things* (rev. ed.). New York: Basic Books.

Norman, D., & Stappers, P. J. (2016). DesignX: Design and complex sociotechnical systems. *Journal of Design, Economics, and Innovation, 1*(2). http://dx.doi.org/10.1016/j.sheji.2016.01.002.

Peirce, C. S., & Burks, A. (Eds.). (1958). *Collected papers of Charles Sanders Peirce* (Vol. 7). Cambridge, MA: Harvard University Press.

Pinet, J. (2015). *Facing the unexpected in flight: Human limitations and interaction with technology in the cockpit*. Boca Raton, FL: CRC Press.

Pinet, J. (2019). Dysfunctions of complex systems: A time-wise method for the analysis of non-linear problems in HSI – Role of expertise and experience. Paper presented at INCOSE 2019 International Conference on Human Systems Integration, Biarritz, France.

Plattner, H., Meinel, C., & Leifer, L. (2016). *Design thinking research: Taking breakthrough innovation home*. Understanding Innovation Series, Cham: Springer International.

Poltrock, S., & Grudin, J. (1994). Computer-supported cooperative work and groupware. Tutorial, *CHI'94 Conference Companion, Boston, MA, and New York: ACM Press*.

Rasmussen, J. (1983). Skills, rules, knowledge: Signals, signs and symbols, and other distinctions in human performance models. *IEEE Transactions on Systems, Man and Cybernetics, 13*, 257–266.

Rochlis Zumbado, J. (2015). *Human Systems Integration (HSI) practitioner's guide*. NASA/SP–2015-3709. Lyndon B. Johnson Space Center Houston, Texas 77058. November. http://ston.jsc.nasa.gov/collections/TRS.

Rothrock, L., & Narayanan, S. (Eds.). (2011). *Human-in-the-loop simulations: Methods and practice*. Berlin: Springer.

Sarter, N. B., Woods, D. D., & Billings, C. E. (1997). Automation surprises. In G. Salvendy (Ed.), *Handbook of human factors and ergonomics* (pp. 1926–1943). New York: Wiley.

Shneiderman, B. (1986). *Designing the user interface: Strategies for effective human–computer interaction,* Reading, MA: Addison-Wesley.

Shneiderman, B. (2002). *Leonardo's laptop: Human needs and the new computing technologies*. Cambridge, MA: MIT Press.

Sheridan, T. B. (1987). Supervisory control. In G. Salvendy (Ed.), *Handbook of human factors* (pp. 1243–1286). New York: Wiley.

Stanton, N. A., Walker, G. H., Salmon, M., & Hancock, P. A. (2017). State-of-science: situation Awareness in individuals, teams and systems. *Ergonomics, January.* DOI: 10.1080/00140139.2017.1278796.

Suchman, L. A. (1987). *Plans and situated actions: The problem of human-machine communication*. Cambridge: Cambridge University Press.

Tillman, B., Fitts, D. J., Woodson, W. E., Rose-Sundholm, R., & Tillman, P. (2016). *Human factors and ergonomics design handbook* (3rd ed.). New York: McGraw-Hill Education.

UK MOD Chiefs of Staff (2018). Human-machine teaming. joint concept note (JCN) 1/18, May. Swindon, Wiltshire: DCDC, Ministry of Defence.

Vega-Mejia, C. A., Montoya-Torres, J. R., & Islam, S. M. N. (2016). Classification of economic, environmental and social factors in vehicle loading and routing operations. Paper presented at ILS2016 - Information Systems, Logistics and Supply Chain 6th International Conference, Bordeaux, France.

Weiser, M. (1991). The computers for 21st century. *Scientific American, Special Issue on Communications, Computers, and Networks.*

Wickens, C. D., & Sebok, A. (2013). Flight deck models of workload and multi-tasking: An overview of validation. In *Proceedings of the 17th International Symposium on Aviation Psychology* (pp. 189–194). https://corescholar.libraries.wright.edu/isap_2013/82.

Wiener, E. & Nagel, D. (1988). *Human factors in aviation*. Academic Press. eBook.