



HAL
open science

From Rigid to Flexible -From Virtual to Tangible An Evolution of Human-Centered Design

Guy Andre Boy

► **To cite this version:**

Guy Andre Boy. From Rigid to Flexible -From Virtual to Tangible An Evolution of Human-Centered Design. Proceedings of 2018 World Congress of the International Ergonomics Association, 2018. hal-02956587

HAL Id: hal-02956587

<https://hal.science/hal-02956587>

Submitted on 3 Oct 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

From Rigid to Flexible – From Virtual to Tangible An Evolution of Human-Centered Design

Guy André Boy^{1,2}

¹ Air and Space Academy, 1 avenue Camille Flamarion, 31500 Toulouse

² ESTIA, Technopole Izarbel, 92 allées Théodore Monod, 64210 Bidart, France
guy.andre.boy@gmail.com

Abstract. Human Centered Design (HCD) has become a necessary and unavoidable approach to seriously consider human factors upstream in systems architecture and functionalities. 20th century practices started by inventing and building tangible objects, functionalities being added incrementally and piled up at infinity, offering not only more automated systems but also more complex uses of these systems. Conversely, since the beginning of the 21st century engineering projects are designed from a computer (i.e., in a virtual environment) by defining scenarios and functional configurations that can be tested using human-in-the-loop simulations where the issue of tangibility is becoming crucial along three dimensions: technology, organizations and people (jobs). These virtual structures and functions must be made tangible from two points of view: that of physics and that of the figurative (i.e., cognitive and socio-cognitive). Tangibility can be characterized and evaluated through five dimensions: complexity; maturity; flexibility; stability; and sustainability. It is interesting to note that these dimensions can be mirrored with that of autonomy: inter-connectivity, independence, flexibility, resilience, and persistence. In this perspective, this article presents a new paradigm, the Human-Systems Integration (HSI) and analyzes the evolution of rigid automation towards a flexible autonomy, proposing a new paradigm of HCD.

Keywords: Human-Centered Design, Flexibility, Tangibility.

1 Introduction

Fig. 1 presents the evolution of three scientific and technical fields of investigations: Human Factors and Ergonomics (HFE); then Human-Computer Interaction (HCI); and today Human-Systems Integration (HSI) as an association of Human-Centered Design (HCD) and Systems Engineering (SE) (Boy & Narkevicius, 2013).

Let's make a distinction between task (what should be done, i.e., a prescription) and activity (what is effectively done). HFE was developed since the end of the World War 2, based on the evaluation of end products, leading to corrective ergonomics. Indeed, HFE is commonly centered on activity analysis of existing or entirely developed systems, and not or very little of systems being designed. Consequently, it is difficult and often impossible to consider HCD recommendations because they arrive too late.

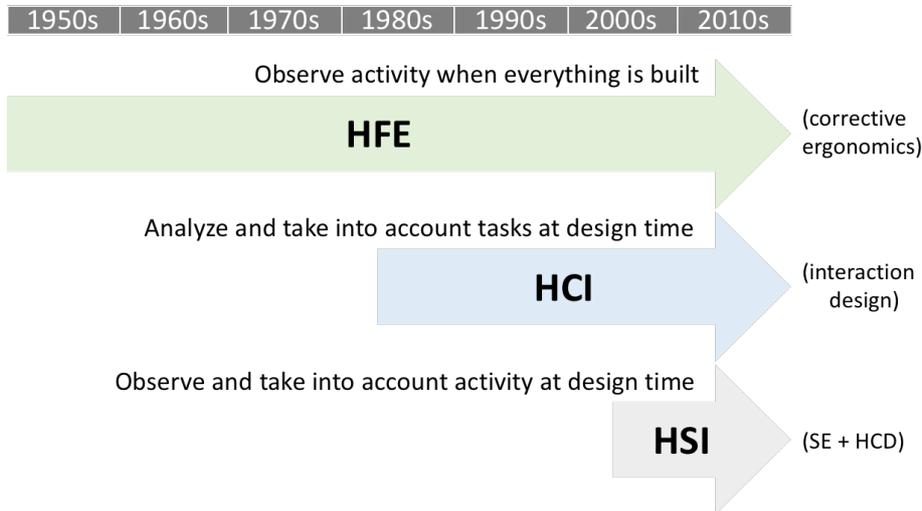


Fig. 1. Evolution from HFE to HCI to HSI.

From the beginning of the 1980s, computer scientists, cognitive scientists and designers developed HCI. Most of related work emphasized usability of computers and interaction design. Most HCI developers are interested in designing user interfaces that enable easy interaction with software applications. HCI contributed to produce methods and tools based on task analysis for HCD, but not too much on activity analysis. This is due to the fact that, using computing systems, tasks and activities are very close (e.g., there is no discrepancy between a prescribed task and its related effective activity on desktop user interfaces). Success of office-automation HCI led the aeronautics community to develop cockpit instruments, introducing the concept of “interactive cockpit”, which denotes the use of pointing devices and graphical user interfaces on flight decks.

HSI results from an evolution of HFE and HCI, associated with SE (i.e., human-centered design and development of complex systems). We had to wait the beginning of the 2000s to start a new trend within SE to concretely emphasize HSI. Modeling and simulation software started to enable Human-In-The-Loop Simulation (HITLS) and therefore to consider human operator’s activity in the design of complex systems. Indeed, activity observation and analysis became possible, and consequently HCD became possible by enabling cognitive/physical function analysis. HSI now includes methods and tools from HFE and HCI during the design process and all along the life cycle of a product.

2 Human-In-The-Loop Simulation for Human-Centered Design

Modeling and simulation in engineering design, often synthesized as “virtual engineering,” enable design teams to carry out agile HCD and development. System functions and structures can be incrementally defined and tested very early during the design

process enabling HSI. This approach contributes to avoiding most automation surprises at operations time (Sarter, Woods & Billings, 1997). It emphasizes flexibility of development processes and tangible solutions.

During the 20th century, engineers designed and manufactured machines that could not be tested before the entire system was built and assembled, leading to a corrective instead of a proactive approach. Systems could not be modified in depth when necessary mostly because heavy investments and commitments were made. This led to the development of user interfaces, operational procedures and users' guides that contributed to adapt people to machines and sometimes compensate design flaws. Consequently, what should we do? How HCD could help considering human stakeholders in the design process to optimize HSI.

In order to answer these questions, let's take National Aeronautics and Space Administration (NASA) lifecycle phases as a support for explaining how HCD works compared to traditional technology-centered engineering (Table 1).

Table 1. NASA lifecycle phases.

	Technology-centered	Human-centered
Pre-Phase A	Concept, Studies	Feasible concepts, simulations, studies, models, mockups
Phase A	Concept and Technology Development	Concept definition, simulations, analysis, models, trades
Phase B	Preliminary Design & Technology Completion	Mockups, study results, specifications, interfaces, prototypes
Phase C	Final Design, and Fabrication	Detailed designs, fabrication, software development
Phase D	System Assembly, Integration and Test, Launch	Operations-ready system with related enabling products
Phase E, F	Operations and Sustainment, Closeout	

Technology-centered approach: Fig. 2 presents three trends (resource commitments; design flexibility and system knowledge) with respect to these lifecycle phases. Resource commitments are quickly growing in the beginning, causing a quick drop of design flexibility. System knowledge grows as system design and development project makes progress. However, it grows slowly in the beginning and grows faster and higher in the end of the lifecycle.

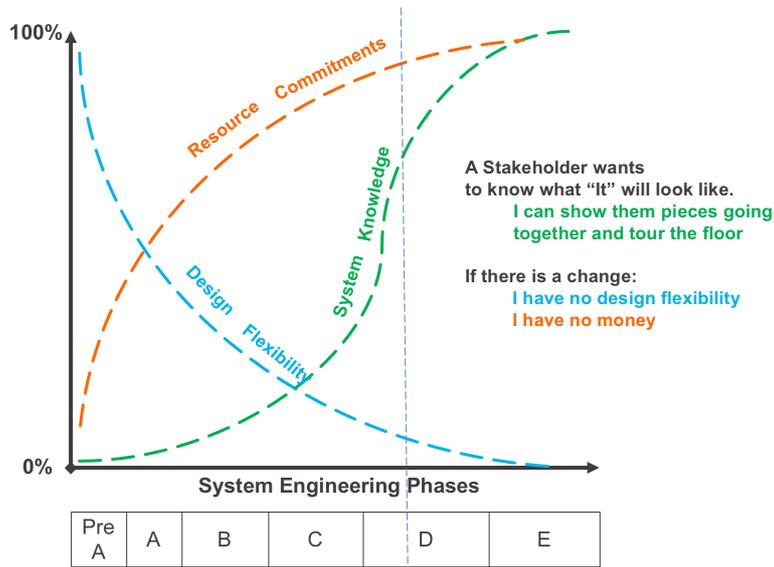


Fig. 2. Late in the lifecycle.

Human-centered approach: Fig. 3, along the same axes, shows that curve concavities are inverted. This is due to upstream use of modeling and simulation, and most importantly HITLS, which provide support for activity observation and analysis that supports HCD.

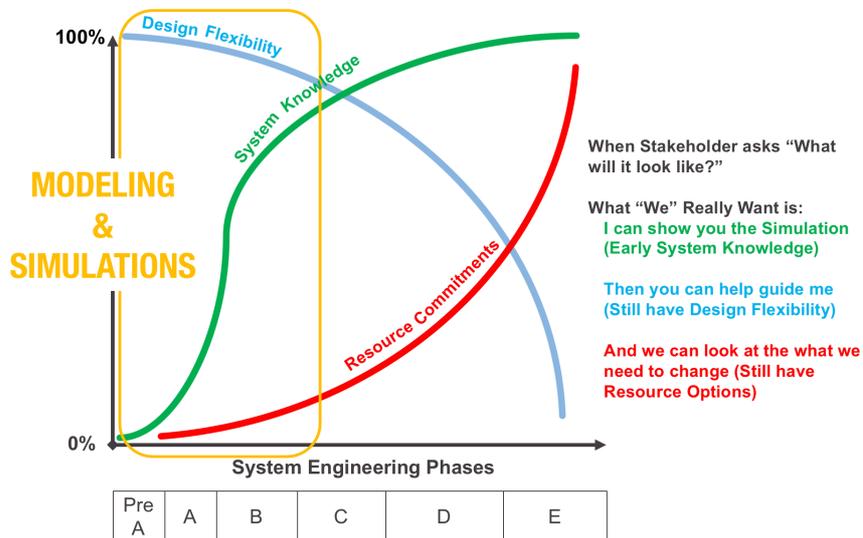


Fig. 3. What we really want (note that "System Knowledge" needs to be interpreted as "Knowledge on Technology and Related Practices").

This approach enables the acquisition of very early HSI knowledge, which is very beneficial for the overall system development. In addition, it is important to keep system flexibility long enough during the lifecycle to modify structures and functions with respect to activities observed in HITLS and further analyzed. Finally, enough flexibility on resource options can be kept long enough.

HCD consists in adopting an architect approach instead of a builder approach where prototypes are incrementally developed and tested using HITLS. During the last three decades, we learned a lot about HCI. More specifically, we learned interaction design from various kinds of task analyses. Making user interfaces, as pieces of software, is now very much mastered, especially as far as usability of concerned. We learned how computers can be used to improve HCD... of computer systems! It is time now to expand this approach to Cyber Physical Systems (CPSs) and Systems of Systems (SoS). In addition to better understanding the making of new jobs while designing new technology, we can also discover or define new kinds of organization setups. HCD is therefore a matter of Technology, Organization and People (the TOP model – Boy, 2013).

3 Human-Systems Integration (HSI) as a new paradigm

At this point, it is important to define what we mean by “system.” Nowadays, a technological system includes software and hardware. Using artificial intelligence terminology, a system is an agent. In Minsky’s sense, an agent can be represented as an agency of agents (Minsky, 1985). In the same way, systems engineering now considers systems of systems (Landauer & Bellman, 1996; Luzeau & Ruault, 2008). Since we are describing HSI, humans and machines can be modeled as interconnected systems or agents in the sense of systems of systems or agencies of agents capable of satisfying one or several objectives (or purposes). I developed the cognitive function model as a triplet (role, context of validity and resources) for both humans and systems (Boy, 1998, 2011, 2013).

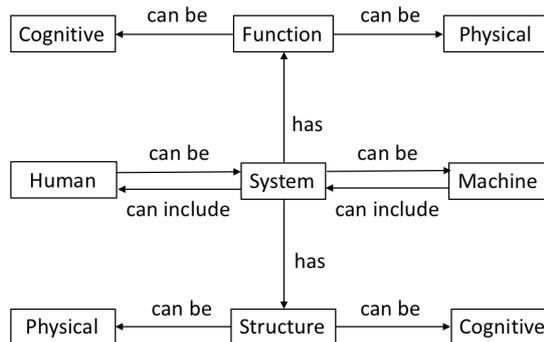


Fig. 4. Humans & machines as systems with cognitive and physical functions & structures. Recursive definition (i.e., humans & machines may include systems).

Both cognitive functions and physical functions can be represented within the same framework (Boy, 2017). When a human uses a system, he or she interacts with this

system. Since current systems have their own cognitive functions, we now talk about human-system teaming (Tangney, 2016). Consequently, within the framework of HSI, a system can be defined as including machines and people. This is a matter of representation (i.e., of course, people are not systems in the technological sense). For example, biologists talk about the pulmonary system when they describe how the human lung is structured and works. In addition, humans and machines can recursively include systems. More generally, we will state that a system has cognitive and physical functions and structures (Fig. 4).

4 From Rigid Automation to Flexible Autonomy

Digital technology enables people and systems to become more automated (proceduralized) and/or autonomous (capable of solving problems), in societal or work-related, civilian or military environments. If this statement is commonly acknowledged, it is important to better understand what we mean by “automation” and “autonomy.” Fig. 5 presents a perspective for identifying the automation-autonomy distinction.

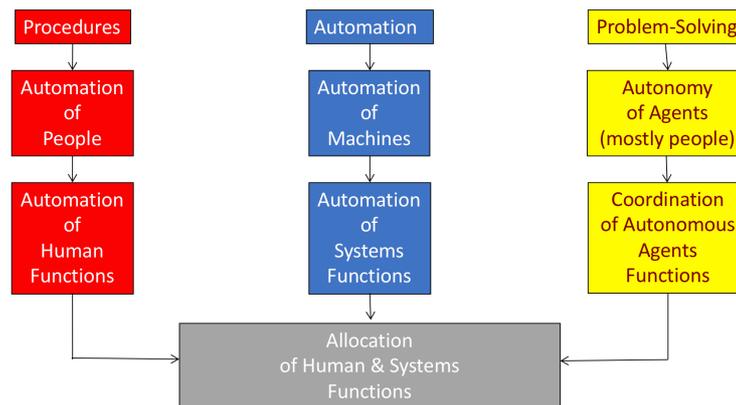


Fig. 5. Automation and autonomy.

We can say that automation results from the use of procedures that enable an agent to be guided in the performance of a task without external assistance. If automation is commonly thought as automation of machines (i.e., automation of systems functions using automatic control theories developed in electronic and mechanical engineering), procedures can be thought as automation of people (i.e., automation of human functions based on paper or electronic procedure following).

Technological automation¹ was implemented using analogic (electronic) devices² until they became software-based (i.e., digital procedures). From a cognitive engineering point of view, both human and machine automations were developed at the skill-based level and rule-based level, in the Rasmussen's sense. When neither behavioral levels worked, human operators had to solve problems by themselves at a higher level of cognition, which has been called the "knowledge-based level" by Jens Rasmussen (1983). The knowledge-based level involves identification of the situation, decision-making and planning of an appropriate set of actions (i.e., make a procedure in real-time). This is also called problem-solving (Fig. 5) that is only successfully possible when the agent is autonomous or can rely on appropriate external help. When there are several autonomous agents, a coordination is required in the form of coordination rules or a supervisor that regulates agents' activity. In all cases, the central problem is to allocate properly human and system functions.

An autonomous system is a highly interconnected system that is able to execute a set of tasks without external help within a defined context. This means that an autonomous agent cannot be designed in isolation from its environment. In other words, a multi-agent approach is required to correctly define autonomy. Multi-agent systems can be clustered into agencies that are themselves agents (i.e., an agent is a society of agents in Minsky's sense). In addition, each agent has at least a cognitive function that is defined by its role, a context of validity (of its autonomy), and associated resources. We call "associated resources" physical and cognitive resources that the system can rely on. An agent has its own learned practices and/or integrated prostheses. More generally, an agent has internal and/or external resources (i.e., an agent has its own resources or can rely on other-agents' help). Simplifying, an agent, considered as an autonomous system, has physical structures and functions (e.g., sensors, effectors), as well as cognitive structures and functions (e.g., information processing and learning capabilities). These structures and functions enable the autonomous system to execute a set of tasks in a given context, which includes space/time contexts, as well as normal, abnormal and emergency contexts. In other words, we define context as context of contexts. It should be noted that the environment of an agent involves both cognitive and physical entities, which typically are an organized set of agents interacting among each other. Three models of autonomy can be defined (Boy, 2013, 2002):

- (1) **Supervision.** When agents do not know each other, the best way to interact safely, efficiently and comfortably is to be supervised. None of the supervised agents has the authority to decide what to do; a supervisor does it for them and tell them what they should do.

¹ Examples of automated systems are thermostats, cruise control systems on cars, and autopilots on aircraft. Automation is achieved using mechanical or electronic devices or software.

² More specifically, automatic control is achieved using feedback controllers developed by electrical and mechanical engineers using control theories. For a long time, automation was replacing skill-based human functions. First autopilots were introduced on commercial aircraft in the 1930s. Later on, flight management systems were introduced on commercial aircraft in the 1980s using operational research, optimization and expert systems. They replaced rule-based human functions.

- (2) **Mediation.** Agents may not know each other but could interact among each other through a common frame of reference (CFR) that mediates their interactions. The CFR can be seen as an organization of mediating agents that facilitate interactions. For example, on WYSIWYG user interfaces, mouse-sensitive desktop metaphors enable easy interaction of people with computers.
- (3) **Cooperation by mutual understanding.** Cooperation by mutual understanding is what people usually do when they interact with each other. This model assumes that agents are able to construct a mental model of the others in order to perform better in future interactions. People interacting among each other do this naturally. Very simple instances of such a model have been developed and used so far on computers. For example, some pieces of software are able to learn user's habits and are able to incrementally provide smart options or suggestions. This is the case of current text processors that are able to learn user's specific lexicon from frequent uses of words. Web browsers remember frequently used links, etc. In this model, authority is traded between the agents.

To summarize, there is a continuum from the supervision model of autonomy where authority follows a top-down army-type model, to the mediation model of interaction where authority follows a transversal orchestra-type model (Boy, 2013), to the cooperation by mutual understanding model of interaction where authority follows a more-chaotic trade model. These interaction models are very useful to support the way cognitive and physical functions are implemented in complex systems not only from a human-computer interaction point of view, but also from system-of-system point of view. In addition, they provide an articulated way to validate autonomous complex systems.

5 From Virtual Prototypes to Tangible Systems

We then need to better understand function allocation among agents, which are systems in the sense of people and machines (see Figure 4), from the beginning of design of highly interconnected systems of systems. Today, any design project starts on a computer (e.g., using PowerPoint, rapid prototyping tools, modeling tools and simulation facilities). HCD cannot be possible without rapid and massive development of digital technologies that contribute to the development digital factories and more generally Industry 4.0.

Unlike during the last century, we start from functions that can be tested in digital environments and end up with Cyber Physical Systems – we then go from software to hardware that already includes software. Good news is that HCD has become possible because we can test, understand and take into account human factors within the entire (modeled and simulated) system from the beginning of the design process (Boy, 2013). More problematic is to find out about system tangibility as early as possible before the system is delivered.

This discussion opens the door to a broader question: what is tangibility? **Tangibility** can be interpreted in two ways: physical and figurative. An example of physical tangibility is when you grasp a physical object. You can touch it, weigh it, push it and so on. An example of figurative tangibility is when you question somebody else about what

he or she is saying. You can or cannot grasp a concept, an abstraction; let's say a figurative object. You may say: "What you say is not tangible!" In fact, you do not understand or believe an argument, an idea, a concept or an abstraction. When we talk about a system, physical tangibility is related to structure, as figurative tangibility is related to function. During the 20th century, mechanical engineering enabled us to develop means, in the form of concrete structures, which themselves enabled us to incrementally develop purposes, in the form of functions developed in virtual environments first, which enable us to deduce implementation means, in the form of tangible structures.

This leads to the concept of Tangible Interactive System (Boy, 2016). HCI already provided many approaches and techniques to handle interaction design (i.e., design and development of software-based interactive systems). Today, 3D printing and additive manufacturing bring new challenges that are related to physical tangibility (i.e., making sure that the system is physically tangible) as well as figurative tangibility (i.e., making sure that the purpose of the system is figuratively tangible). We can integrate various kinds of TISs at design time in virtual environments and test them with people in the loop, but we need to make sure that the resulting system will be tangible in the end.

In HCD, looking for tangibility leads to developing autonomy. Indeed, defining an autonomous system as a hyper-connected entity capable of detecting its environment, making appropriate decisions, acting and consequently learning from its experience. This is the same as increasing familiarity with complexity of the various interconnections, verifying its maturity in terms of robustness and reliability for example, insure flexibility in operations and maintenance, verifying stability in terms of resilience, et insure sustainability. These criteria are the same as those of tangibility (Boy, 2016). Consequently, building an autonomous system requires specific attention in terms of test of involved cognitive and physical functions, as well as their various interconnections. This (analytical and experimental) tangibility investigation in autonomous systems design, made in terms of cyber-physical functions and structures, leads to the identification of new activities, and therefore new jobs, as well as new coordination rules, and therefore new organizations.

6 Conclusion

Current digital technology offers environments that facilitate human- and organization-centered design very early in the life cycle of a system. This HCD evolution enables us to define as correctly as possible functions involved in a socio-technical system incrementally developed and used. We saw that using virtual prototyping tools require considering the tangibility concept seriously. Tangibility will be considered from two viewpoints: physical and figurative (i.e., cognitive or socio-cognitive). Five properties of tangibility will be taken into account: complexity, maturity, flexibility, stability and sustainability (Boy, 2016). These five properties apply to autonomy also. We will investigate the tangibility of a system to deduce its degree of autonomy within its environment.

This article provides a new framework that will enable us to represent autonomous and semi-autonomous complex systems, in the sense of human and machine agents,

interacting among each other. This framework will be used to improve function allocation among the various agents with respect to situations, whether they are normal, abnormal or emergency. Satisfactory socio-technical solutions will be found by: (1) progressive problem formalizations, by using ideation techniques, conceptual model development, rapid prototyping and other scenario-based methods; (2) HITLS involving subject matter experts and using activity analysis methods; (3) functional and structural analyses, by refining functional and structural representations, as well as concepts of role, context and resources; and (4) lots of tests that must be repeated until acceptable solutions will be found.

The right mix between objective information (coming from data analysis) and subjective information (coming from experts and credible people, more specifically operations people in appropriate domains, who can elicit appropriate use cases correctly) should be found in the analysis, design and evaluation of complex systems being developed. Human-centered design of autonomous complex systems should be further defined and developed. This article provides an approach that the FlexTech Chair started to develop in several industrial projects.

References

1. Boy, G.A.: *Cognitive Function Analysis*. Greenwood/Ablex, CT, USA (1998).
2. Boy, G.A.: Theories of Human Cognition: To Better Understand the Co-Adaptation of People and Technology, in Knowledge Management, Organizational Intelligence and Learning, and Complexity, Ed. L. Douglas Kiel, in *Encyclopedia of Life Support Systems* (EOLSS), Developed under the Auspices of the UNESCO, Eolss Publishers, Oxford, UK (2002).
3. Boy, G.A.: *Orchestrating Human-Centered Design*. Springer, UK (2013).
4. Boy, G.A. (2011). (Ed.) *Handbook of Human-Machine Interaction: A Human-Centered Design Approach*. Ashgate, UK.
5. Boy, G.A.: *Tangible Interactive Systems*. Springer, U.K (2016).
6. Boy, G.A. (2017). Human-Centered Design of Complex Systems: An Experience-Based Approach. *Design Science Journal*. Vol. 3, Cambridge University Press, U.K.
7. Boy, G.A. & Narkevicius, J.: Unifying Human Centered Design and Systems Engineering for Human Systems Integration. In *Complex Systems Design and Management*. Aiguier, M., Boulanger, F., Krob, D. & Marchal, C. (Eds.), Springer, U.K. (2013).
8. Landauer, C. & Bellman, K.L.: *Collaborative system engineering and integration environments*. 5th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (1996).
9. Luzeau, D. & Ruault, J.R. Eds.: *Systems of Systems*. John Wiley, Hoboken, NJ, USA (2008).
10. Minsky, M.: *The Society of Mind*. New York: Simon and Schuster (1985).
11. Rasmussen, J.: Skills, rules, and knowledge-signals, signs, and symbols, and other distinctions in human performance models. *IEEE Transactions on Systems, Man, and Cybernetics*, 13(3), pp. 257–266 (1983).
12. Sarter, N.B., Woods, D. D. & Billings, C.E.: Automation surprises. *Handbook of Human Factors & Ergonomics*, second edition, G. Salvendy (Ed.), Wiley, USA (1997).
13. Tangney, J.: Human Systems Roadmap Review. Presentation DCN# 43-1322-16. National Defense industrial Association (NDIA) Human Systems Conference (<https://ndiastorage.blob.core.usgovcloudapi.net/ndia/2016/Human/Agenda.pdf>) (2016).