
# Model-Based Human Systems Integration

Guy André Boy

## Contents



G. A. Boy (✉)
CentraleSupélec, Paris Saclay University, Gif-sur-Yvette, France

ESTIA Institute of Technology, Bidart, France
e-mail: guy-andre.boy@centralesupelec.fr; g.boy@estia.fr



A. Madni et al. (eds.), *Handbook of Model-Based Systems Engineering*,
https://doi.org/10.1007/978-3-030-27486-3_28-1

**Abstract**

Human systems integration (HSI) is an essential field of systems engineering (SE) that emerged, departs, and encompasses from its initial components that are human factors and ergonomics, human-computer interaction, engineering, and domain experience. Current capabilities and maturity of virtual prototyping and human-in-the-loop simulation (HITLS) enable virtual human-centered design (HCD) that can be combined with SE to realize HSI. HSI is almost necessarily model-based; it uses HITLS and requires a homogenized human and machine systemic representation. Virtual HCD enables us to take into account both human and organizational elements not only during the design process but also during the whole life cycle of a system. These new capabilities are made possible by digital tools that enable virtual environments that in turn should be made tangible. Digital twins can be solutions for supporting HSI, operations performance, and experience integration. Tangibility is therefore a crucial concept in model-based HSI (MBHSI), which should be both analytical and experimental, based on appropriate scenarios and performance metrics essentially supported by domain experience. An aeronautical example illustrates an instance of MBHSI.

**Keywords**

Human systems integration · Systems engineering · Human factors and ergonomics · Human-computer interaction · Human-in-the-loop simulation · Modeling · Virtual prototyping

## Introduction

When the first draft of this chapter was started, the intention was to provide a model-based systems engineering (MBSE) contribution to human systems integration (HSI). Thinking about it, it became clear that HSI is necessarily model based. This is the reason why the title of this chapter is Model-Based Human Systems Integration (MB-HSI), which reinforces the intrinsic modeling need for HSI. In addition, modeling in HSI cannot be considered without simulation and even more importantly human-in-the-loop simulation (HITLS).

HSI emerged in the beginning of the 2000s as an approach to considering the human element in the design and management of a complex sociotechnical system during its whole life cycle. A sociotechnical system (also called human-machine system) is composed of humans and machines interacting with each other.

The progressive and exponential accumulation of software during the last three decades naturally induced the need for considering human-centered design (HCD) seriously [9, 11]. More specifically, HCD was made possible thanks to the development of computer-based prototyping that enables "virtual HCD" (VHCD). VHCD involves HITLS that enables carrying out activity analyses and further considers tangibility. Virtual HCD is further developed in section "State of the Art: History and

Evolution" of the chapter. Two tangibility meanings should be articulated in the design and management of complex sociotechnical systems: physical tangibility (e.g., grasping a physical object) and figurative tangibility (e.g., grasping a concept or an abstraction). Today, some tangibility problems can be anticipated through the use of 3D printing capabilities, for example.

Eugène Ionesco, a French avant-garde theater writer, depicted human existence in a tangible way. Claude Bonnefoy published, after Ionesco's death, a magnificent interview he had with him, where the following citation is an excellent perspective for VHCD: "In a dream, we're still in a situation. In short, I believe that dreaming is both a lucid thought, more lucid than in the waking state, a thought in images, and that it is already theatre, that it is always a drama because we are always in a situation (French citation: "*En rêve, on est toujours en situation. Bref, je crois que le rêve est à la fois une pensée lucide, plus lucide qu'à l'état de veille, une pensée en images et qu'il est déjà du théâtre, qu'il est toujours un drame puisqu'on y est toujours en situation.*")" [32]. Virtual prototyping and advanced visualization techniques and tools provide us with means that transform dreams into images, theater plays, and consequently tangible designs. Brenda Laurel introduced this concept in her book, *Computer as Theater*, in the beginning of the 1990s [37].

This evolution toward a human-centered systemic approach requires to define what the concept of "system" really means. Indeed, the concept of system is not only a matter of machines, but it is also a matter of people and organizations. This is the content of section "Key Concepts and Definitions for a Human-Centered Systemic Approach" of the chapter, where a system is defined as an articulation of structures and functions and issues of complexity and maturity of human-machine systems are developed.

Section "Domain Experience Integration and Artificial Intelligence Solutions" is an attempt of clarification of the concept of experience, as well as its utility in HSI. A systems engineering framework for model-based experience integration is provided. This framework could be based on appropriate artificial intelligence (AI) concepts, methods, and tools. Domain experience is thought in the phenomenology sense (i.e., meaning and subjectivity of people's experience from observation and compiled knowledge in a given domain).

Section "Concrete Chapter Contribution: The PRODEC Method" presents a new method, called PRODEC, based on the acquisition, analysis, and use of procedural (PRO) and declarative (DEC) knowledge for VHCD. Human-centered design of complex systems is a matter of identification of relevant human and machine entities, considered as systems, which can be physical and/or cognitive (cyber). Procedural knowledge is about operational experience that is often expressed in the form of stories by subject matter experts. Declarative knowledge is about objects and agents involved in the targeted human-machine system being designed. The PRODEC process involves human-in-the-loop simulation to incrementally create and maintain appropriate performance models. An aeronautical application is provided to illustrate the use of the PRODEC method.

A discussion is started in section "Discussion: Challenges, Gaps, and Possible Futures" where the approach departs from technology-centered MBSE [31] toward

human-centered design intimately fused into systems engineering (SE) that naturally leads to human systems integration. Human-centered modeling limitations are explored leading to an epistemological endeavor.

The summary provides perspectives in model-based human systems integration where possible directions of research are projected not only on human modeling but also on human-centered digital twins.

## State of the Art: History and Evolution

### Task and Activity

It is useful to introduce the distinction between task and activity. A task is what is prescribed to be performed. For example, preparing breakfast in the morning is a task. It can be described in the form of subtasks, including "taking fruits out of the fridge," "putting cereals and soymilk in a bowl," "preparing coffee," and so on. It becomes clear that a task is linguistically described by a verb and a direct object complement. A task can also be described by a procedure, which can be a list of subtasks or a more sophisticated algorithm of subtasks, like a computer program that includes sequences, choices, loops, and other things of that type. Flying an aircraft is based on procedure following, for example. If a task is usually prescribed before actual performance using a procedure, this procedure can be occasionally adapted. For example, when breakfast becomes a brunch, other ingredients could be added to the everyday routine, and you can make another more elaborated procedure for that occasion. In terms of task, you are always brought back to follow a procedure, whether already made or made on the spot.

However, at operations time, things may not happen exactly as expected. There are events or circumstances that may disturb the execution of the task. In these cases, procedures do not apply as prescribed, and human operators need to make something up. Consequently, their actual activity departs from the task (i.e., what you do effectively is different from what was prescribed). For example, you realize that you forgot to buy coffee, and the subtask "preparing coffee" cannot be executed as prescribed. At this point, you have to solve a problem. You may decide to choose the task "preparing tea," or "going to the shop, if open, and buy coffee." Activity could be called effective task that is an adaptation of the task to the circumstances in real time. The expression, "deviations between task and activity," is typically used.

In life-critical environments, procedures greatly support operations (i.e., performance), but problem-solving skills and knowledge are also necessary to handle unanticipated circumstances or unexpected events. Reasons of task-activity deviations can be intrinsic (e.g., a human operator suddenly becomes sick) or extrinsic (e.g., weather conditions cause an electric shutdown). In both cases, adaptation is required and effectively happens. This is the reason why it is essential to anticipate possible activities in human systems integration. This has been a major difficulty for a long time. The next section presents an evolution of engineering and

associated human factors disciplines that contributed to bring solutions to this task-activity issue.

## Evolution of Engineering and Associated Human Factors

The evolution of human-related engineering approaches can be decomposed into three eras (Fig. 1). Prior to the 1980s, the most important disciplines in engineering were mathematics, physics, mechanical engineering, and industrial engineering. This era was dominated by hardware and human physical issues at work. Human factors and ergonomics (HFE) started to develop after World War 2 as an evaluation discipline, populated with occupational medicine doctors, caring about workers' health and safety. Health at work dealt with physical injuries, humidity, noise, heavy physical workload, and so on. Safety was in terms of physical danger. When new systems were built, people's activity had to be observed and analyzed before design and after the complete development of the system. This was fine, except for the fact that human factors specialists were always fighting with engineers to convince them that significant sociotechnical changes should be done on the system developed. They were always too late, after design and development processes were over and budget was almost all spent.

The second period started with the development of personal computers during the 1980s and later on with the Internet. Universal access to computing grew exponentially. Software engineering, computer graphics, AI, and cognitive engineering developed very rapidly. Software and computing were implemented in the form of

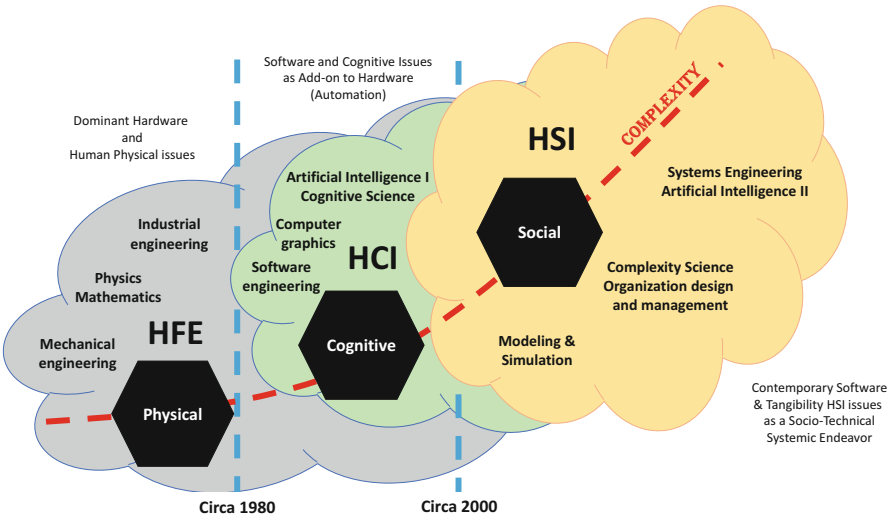


**Fig. 1** Evolution engineering and associated human factors

add-ons to hardware. This is what automation is about (e.g., automation of airplanes, office automation, and so on). People incrementally discovered new cognitive issues that were unknown up to then. The shift was from doing to thinking. Human-computer interaction (HCI) developed as a discipline, making graphical user interfaces (GUIs) a necessity. Usability engineering became a dominant practice in HCI, because it became possible to test a system with end users in the loop. HCI developed because both fundamental and practical supports for interaction design and task analysis were needed as a major technique in HCD. Interaction design is very contextual (i.e., user experience is crucial to HCD, and tests should be performed in a large number of contexts). HCI provided what is available today in terms of GUIs, data and information visualization, computer-supported cooperative work (CSCW), and smartphones, for example. Commercial aircraft flight decks that increasingly include layers and layers of software and computer-based pointing devices were called "interactive cockpits." Pilots started to interact with computers more than with mechanical devices of their aircraft.

HCD first focused on usefulness and usability engineering [46] and the development of ISO 9241-210:2010(E) standards, for example. At the same time, even if participatory design was developed within the HCI community [45], it should be recognized that aeronautical engineering design was for a long time, and still is, based on participatory work with experimental test pilots. Participatory design requires everyone to understand each other and more specifically the definition of a common language, based on the various concepts of the domain. It should be evolutionary because it is impossible to get a definitive ontology but a stabilized explicit ontology. This language should support analysis, design, and evaluation of complex systems.

The importance of complexity science in SE, virtual modeling and simulation (M&S), and organization design and management became effectively clear in the beginning of the 2000s. Human systems integration (HSI) was born as a combination of HCD and SE. The problem became less about automation since any technological project is starting nowadays on a computer, using PowerPoint, for example, to present how the future system will look like and it could be used. The next step is typically a more detailed virtual M&S of that system, even human-in-the-loop simulation (HITLS) where people, potentially end users, have the possibility to test the future system being developed, in a virtual way. However, even if this is great news (i.e., activity can be tested at design and development time), the problem of tangibility remains.

Current sociotechnical systems are complex because they are extremely interconnected, and what people's roles are or should be within these systems should be figured out. Interconnectivity is not only between us and technology but also between us through technology. It is therefore important and timely to improve our understanding on what these systems are really about. In addition, since AI is back, how AI and SE can be harmonized and work together should be better understood. Why? This is because systems are becoming more autonomous and therefore have to be more appropriately coordinated. Automation has been developed using knowledge of very well-known situations. Autonomy is currently seen as the next

technological step. It consists in developing systems that are equipped with enhanced situation awareness, decision-making, planning, and action capabilities. However, it would be better to focus more on human autonomy as well as organizational autonomy by developing technology and organizational setups that provide flexibility, instead of current automation rigidity. Therefore, working on human and machine autonomy is a must to better understand how to handle often-forgotten non-linearities.

## System Knowledge Impacts Design Flexibility and Resource Management

System knowledge, design flexibility, and resource commitments are three parameters that should be followed carefully during the whole life cycle of a system. HSI aims to increase the following sufficiently early [6]:

- System knowledge, that is, knowing about systems at design, development, operations, and disposal times, how the overall system, including people and machines, works and behaves.
- Design flexibility, that is, keeping enough flexibility for systems changes later in development and usages.
- Resource commitments, that is, committing as late as possible on expensive resources (e.g., hardware) during the life cycle of the overall system.

When a technology-centered approach is used (typically what has been done up to now), system knowledge increases slowly in the beginning, growing faster toward the end of the life cycle (i.e., from early design to disposal). Design flexibility drops very rapidly, leaving very few alternatives for changes, because resource commitments were too drastic and too early during design and development processes.

Observing the way systems are being designed today helps realize the shift from the twentieth century, where the engineering design process was from hardware to software with constant automation during the last three decades, to the twenty-first century, where the engineering design process is from software to hardware since all designs start on a computer and eventually end up being 3D printed. This shift is good news because there is no need to commit too early on hard-to-modify resources. At the same time, the engineering design team can learn about the system being designed from its digital twin that can be used and then tested (i.e., using HITLS). This means that activity analysis can be done at design time, and results can be injected into the requirements of the system before it is physically built. Even better, enough design flexibility can be kept for a longer period of time.

This drastic revolution makes emerge issues of tangibility that need to be addressed seriously. Tangibility is not only a matter of physics; it is also a matter of intersubjectivity (i.e., mutual understanding) between end users and designers and ultimately between end users and developed technology. Note that in human-machine systems, machines are the concrete realization of designers; this is the

reason why user-designer intersubjectivity is at stake. In other words, end users should be able to understand what machines are doing at appropriate levels of granularity and why. Consequently, complexity analysis has become tremendously important in our increasingly interconnected world. This is even more significant when machines include AI software that provides behaviors difficult to understand and therefore requires explainability mechanisms to keep trust and collaboration with the people involved.

At this point, the need for virtual M&S for HCD and consequently HSI becomes obvious. Virtual M&S is mandatory to develop and use HITLS for making virtual HCD possible (i.e., enabling activity observation and analysis in a virtual world that resembles reality and therefore getting the right system for the right task in the right context). Tangibility tests should be done.

## Use of Digital Twins During System's Whole Life Cycle

The digital twin concept was presented to the industry in 2002 under the term of "Conceptual Ideal for Product Lifecycle Management (PLM)" at the University of Michigan (Grieves, 2016). NASA adopted the term "digital twin" [17, 25, 47, 62]. A digital twin is "a set of virtual information constructs that fully describes a potential or actual physical manufactured product from the micro atomic level to the macro geometrical level" [26]. This definition usually refers to product's structure. It should be extended to product's function. Figure 2 presents a concept map of the digital twin concept.

A digital twin (DT) is a virtual instance of a physical/cognitive system that enables to simulate dynamic phenomena of both structures and functions of a system. Madni et al. (2019) claimed that DT extends MBSE. According to Madni and his colleagues [40], a digital twin can be used as a model of a real-world system to represent and simulate its structure, performance (i.e., function), health status, and mission-specific characteristics during the whole life cycle of the system and incrementally update it from experience (e.g., malfunctions experienced, maintenance, and repair history). In other words, a digital twin can be used as a recipient of experience feedback information and support for system performance (e.g., preventive and timely maintenance based on knowledge of the system's maintenance history and observed system behavior). A digital twin is therefore a great support to improve understanding of the various relationships between system design and usages. In addition, a digital twin enables to support traceability and logistics along the whole life cycle of a system.

Considering a digital twin as a system digital model, in the SE sense, a distinction should be made between predictive digital DT and explanatory DT. A predictive DT is typically a very well-tested digital analog that produces similar outputs as the system would produce in response to the same inputs. It is usually simple and defined in a limited context. It is consequently short-term, rigid, and focused on a specific process or phenomenon. It can be used for marketing and, in some specific cases, when the domain is sufficiently mastered to operationally predict crucial


**Fig. 2** Digital twin definitions and properties

system's states. In contrast, an explanatory DT is defined by an ontology of the domain. It is longer-term, flexible, and generic within the domain being considered. It can be used for analysis, design, and evaluation of a complex system, as well as for documenting its design and development process and its evolutionary solutions. Therefore, a digital twin is a digital model that enables running simulations to predict behavior and performance of a real-world system and/or explain why this system behaves the way it behaves.

A digital twin could be considered as a sophisticated interactive notebook that provides a vivid representation of the system being considered. Validation and certification of a DT are never finished. A DT is constantly modified by integrating new features, as well as modifying and/or removing old features. Considering system's life cycle, digital twins can be used in a variety of industrial activities including product design; engineering optimization; smart manufacturing; job-shop; scheduling; human-machine collaboration; operations diagnostic and decision-making; prognostics and health management; maintenance management; and, more generally, product lifecycle data management.

A DT could be used as a mediating tool to collaboratively test a very early concept within a design team that includes a group of experts with different backgrounds.

Note that the design team includes targeted users or human operators of the system to be developed. Each member of the design team should understand the same thing as the others. This is the reason why team members should have the same objectives and share the same situation awareness (SA) of what is being designed and further developed [22]. In this case, the DT represents this shared SA (SSA). Each member of the design team can see the same thing and eventually manipulate it. It is therefore a great support for participatory design. A DT for SSA starts with a discount DT (DDT). This could be done with the help of an artist, capable of producing a DT in the form of a cartoon or animation of the targeted system to be developed. DDT increases design team intersubjectivity through incremental modifications using collective critical thinking and experience feedback.

Once the DT is fully completed and approved by all members of the design team, it can be used to develop computer-aided design (CAD) of the system in depth. Using real numbers determining system's structure and function contributes to tangibilize the whole thing. The DT then becomes more rational and tangible. Once a "final" version is approved, physical construction of the system can start. This is another stage of tangibilization. Once a satisfactory version of the physical prototype is constructed, it can be tested in the "real world." Nevertheless, handling this HCD participatory process toward HSI requires everyone to speak the same systemic language. Consequently, the concept of system should be clarified in light of HSI.

## Key Concepts and Definitions for a Human-Centered Systemic Approach

### What Does "System" Really Mean?

A system is a representation of a natural or artificial entity. For example, physicians talk about cardiovascular or neural systems; anthropologists talk about communities of people and social groups as organizational systems; and engineers talk about mechanical and civil engineering systems. Figure 3 presents a synthetic view of what

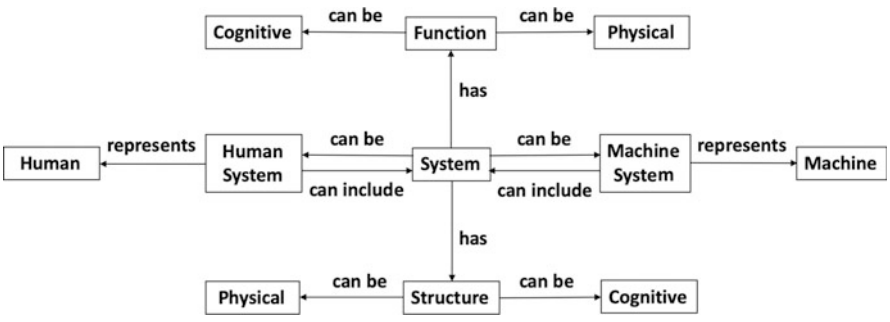


**Fig. 3** Cognitive-physical structure-function system representation

a system is about. A system is recursively defined as including people, machines, and systems. A system can then be considered as a system of systems (SoS). In addition, a system has at least a structure and a function that can be physical and/or cognitive. In practice, a system has several structures and several functions articulated within structures of structures and functions of functions.

At this point, let's notice that an SoS is defined in the same way as Minsky [44] defined an agent as a society of agents. Russell and Norvig [55] defined an agent as an architecture (i.e., structure) and a program (i.e., function). For a long time, engineers considered a system as an isolated system or a quasi-isolated system. As for an agent in AI, which has sensors and actuators, a system in SE has sensors to acquire an input and actuators to produce an output. In an SoS, each system is interconnected to other systems either statically (in terms of systems' structures) or dynamically (in terms of systems' functions). Summarizing, an SoS is projected onto a structure of structures, usually called an infrastructure, where a network of functions could be allocated. It should be noted that the resulting network of functions is not necessarily a direct mapping on the related infrastructure.

The definition of a system is intrinsically recursive (Fig. 4), as an agent is defined as a society of agents in AI [6]. Therefore, in this chapter, "system" and "agent" are representations that have the same meaning. A system's function is defined by a role, a context of validity, and resources that can be physical and/or cognitive, human or machine systems, or agents. In addition, concepts of system and resource are very similar. A resource of a system is a system itself.

Complexity generated by several levels of recursion, shown on Fig. 4, has a direct impact on engineering design and validation of systems being developed. If resources are rigidly allocated to a system, when external context does not match the context of validity of the system, serious issues may arise at operations time. Alternatives are dynamic resource allocation either by adapting existing resources to new jobs or creating new resources. This will be further analyzed later on in this chapter. Resources and contexts are orthogonal and should be articulated. From a methodological point of view, a context-resource hyperspace can be a very useful

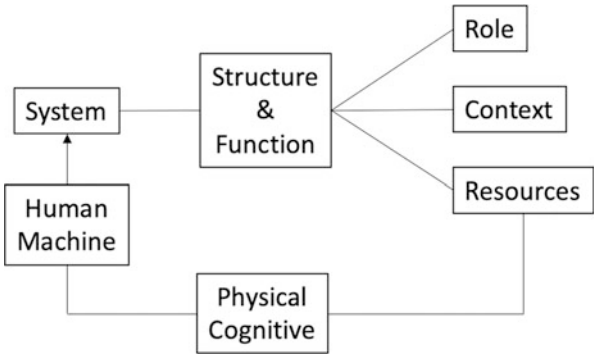


**Fig. 4** Recursive definition of a system

topological support in engineering design and SE. Figure 5 presents a functional context-resource hyperspace that can be mapped onto a dual structural hyperspace.

## Emergent Functions and Structures

Following up on the task-activity distinction, in addition to the topological and teleological definition of a function in terms of role, context, and resources, it is interesting to also consider a function in a logical sense. That is, a function transforms a task into an activity. In other words, when a human-machine system is at work, it produces various kinds of activity resulting from the execution of the various tasks that it is assigned to do. Bertalanffy [5, 58] said "a system is a set of elements in interaction." In addition, a system at work does not stay the same during its lifetime. It learns. Such learning is a matter of incorporation of emergent behaviors and properties. Emergence comes from activity.

Figure 6 shows emerging functions in yellow (i.e., functions coming from problem-solving of unanticipated issues and that should be compiled and incorporated in future practice) and potentially emergent structures in pink (i.e., structures that were previously ignored and need to be incorporated in the design of the overall system).

## Looking for Separability, Emergence, and Maturity

There are three important factors that characterize sociotechnical systems: separability, emergence, and maturity. Physiologists are aware and use the separability concept for a long time to denote the possibility of separating momentarily an organ from the human body without irreversibly damaging the entire human body, considered as a system of systems. Some organs, such as the brain, cannot be separated because the human being could die from this separation. Those organs, as systems, have to be investigated and treated while connected to the rest of the body.

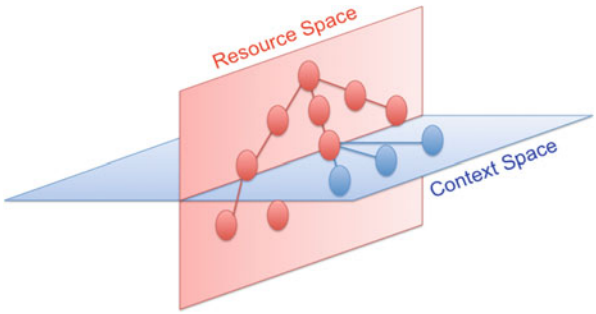


**Fig. 5** A functional context-resource hyperspace (adapted from Boy, 2011)

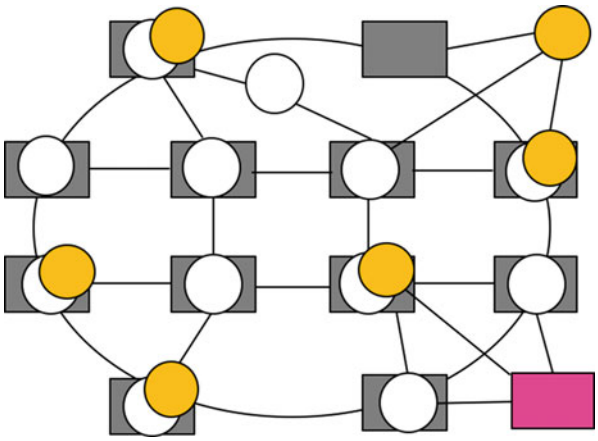

**Fig. 6** Emerging functions (yellow) and structures (pink) within an active human and machine system of systems (note that functions are represented by circles and structures by rectangles)

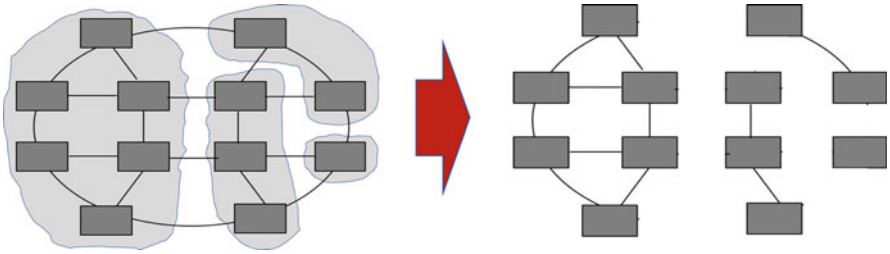

**Fig. 7** Example of three separable systems of a SoS (Boy, 2020)

It is therefore crucial to depart from the system design approach in silos and integration just before system delivery. This is not a problem when sub-systems are separable (Fig. 7). Clumsy integration, often done too late in the development process, is likely to cause surprises and even a few catastrophes at operations time. This is the reason why adjustments are always required, operationally either via adapted procedures or human-machine interfaces and in the worst case more drastic redesign of the whole system.

The lack of consideration for the separability issue in air traffic management (ATM) is a good example, where, for a long time, most air and ground technologies were designed and developed in isolation. Recent programs, such as SESAR (Single European Sky ATM Research), try to associate air and ground stakeholders. The "TOP model" (shown on Fig. 8 and developed in the next section of the chapter) supports design and development teams in the rationalization of interdependencies between technology, organizations, and people [9]. This requires observing and analyzing various activities using virtual M&S to identify emerging properties and functions of technologies under development, and not await to discover them at operations time.

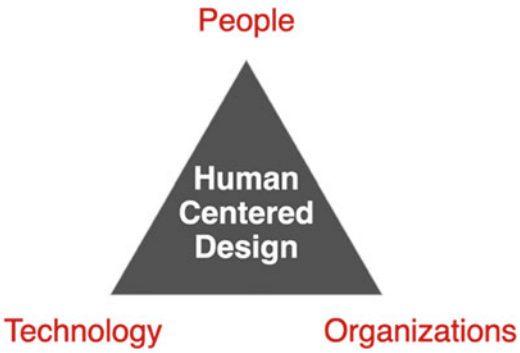


**Fig. 8** The TOP model for HCD

## Domain Experience Integration and Artificial Intelligence Solutions

There is a lot to say on experience feedback, also called in-service experience or user experience depending on the domain at stake (e.g., aeronautical, nuclear, or computer industry). In life-critical systems, experience feedback, often known as REX or RETEX, is cumulative and happens to be heavy duty but contributes to create and maintain a safety culture, for example. More specifically, ultra-safe industries [2], such as the nuclear industry, produce procedures and rules that have become numerous and can constrain operations [13]. This is mainly due to cumulative experience feedback mechanisms that contribute to pile up large numbers of regulatory-based requirements, which end up in such numerous procedures and rules. In human-computer interaction, user experience, often known as UX, is used to refine user interfaces with respect to activity observation and analysis. This section is an attempt to rationalize what experience integration means. But first, let us define the term "experience."

### What Does "Experience" Mean?

The French philosopher and sociologist Auguste Comte introduced positivism that considers authentic knowledge as based on sense experience and positive verification [19]. The German philosopher Edmond Husserl introduced phenomenology in the beginning of the twentieth century as the study of consciousness and conscious experience [29, 30]. Among the most important processes studied by phenomenology are intentionality, intuition, evidence, empathy, and intersubjectivity. The positivism-phenomenology distinction opens the debate on objectivity and subjectivity. Our occidental world based most of our engineering approaches on positivism which led to developing a very precise and verifiable syntax, often leaving semantics somewhere behind, perhaps because semantics is full of subjectivity.

Winograd and Flores [65] provided a perspective for AI and HCI based on phenomenology. This chapter extends this perspective to engineering design and SE. Therefore, referring to phenomenology, the concept of experience that will be used in HSI is about meaning and subjectivity coming from people's experience in a given work environment. Gathered people's experiences will help the construction of typical episodes or scenarios. HSI considers that the classical positivist approach is no longer sufficient, often ineffective, and inappropriate, leaving aside crucial non-linearities that come back to us at operations time in the form of what is currently called "unexpected events" (e.g., COVID-19 pandemic). Experience is gathered and integrated incrementally during (sometimes long) periods of time. Typical episodes are assimilated and accommodated in the form of schemas, in Piaget sense [49, 50]. Piaget's schemas can be represented by cases in AI and lead to case-based reasoning.

Complex human-machine systems are living entities where "normal" events are experienced and/or observed, and emergent phenomena are incrementally discovered; they all are reported as experience. Good human-centered design should focus on the discovery of such emerging phenomena, in HITLS during design and development and in-service experience during the whole life cycle of a system.

## Toward Model-Based Experience Integration: Human-AI-SE Cross-fertilization

Cross-fertilization of SE and AI has been already mentioned in this chapter when the analogy between system and agent was described and more specifically the analogy between a system as a system of systems in SE and an agent as a society of agents in AI [7]. The Research Council of the Systems Engineering Research Center (SERC) and a US Defense Department-sponsored University Affiliated Research Center (UARC) recently developed a roadmap structuring and guiding AI and autonomy research [43]. This roadmap outlines "digital engineering transformation aspects both enabling traditional SE practice automation (AI4SE) and encourage new SE practices supporting a new wave of automated, adaptive, and learning systems (SE4AI)." Even if this roadmap mentions that automation and human interaction research (denoted as manned/unmanned teaming) was "an essential part of systems engineering of these systems," nothing was said on neither organizational issues and problems to be solved nor how human-systems integration would be done. This is the reason why HSI takes even more importance in our growing digital world where autonomy and flexibility have become essential [6].

Why is the shift from automation to autonomy crucial in HSI? Control and management of life-critical systems are typically supported by operations procedures and automation. Automation is usually thought as automation of machines. Analogously, operation procedures can be thought as automation of people [9]. Problems come when unexpected situations occur, and rigid assistance (i.e., procedures and automation) may not work any longer, because system's activity is out of its context of validity. Outside system's context of validity, the rigidity of both procedures and

automation rapidly leads to instability and sometimes radical breakdowns unfortunately. In these cases, instead of following procedures and monitoring automation, people need autonomy to solve problems. The more people have appropriate knowledge and experience, the more they are autonomous and have the capabilities to solve problems. They also need to have appropriate technological and/or organizational support. Therefore, autonomy is a matter of appropriate technological support enabling flexibility, coordinated organizational support, and people's knowledge and knowhow. Off-nominal situations management involves functions of autonomous human and machine agents that need to be coordinated. Figure 9 presents these three options, which lead to the difficult problem of function allocation.

In all cases, functions from automated or autonomous systems/agents need to be correctly allocated to the right systems/agents whether they are people or machines. Such allocation cannot be done only statically a priori but also dynamically with the evolution of context. Consequently, systems should be flexible enough to be able to be modified incrementally. Flexibility should result from such function allocation based on the TOP model.

## Coordinating Technology, Organization, and People (TOP)

On the technology side, machine functions should be flexible enough to be modified if required and appropriately usable. HSI requires integration of experience and expertise, which is often available in the form of cases solved in the past and potentially reusable in similar situations. AI extensively developed knowledge-based systems and case-based reasoning, which can be very effective when associated with supervised machine learning. Handling cases requires appropriate situation awareness, and therefore AI can supply approaches such as intelligent visualization,

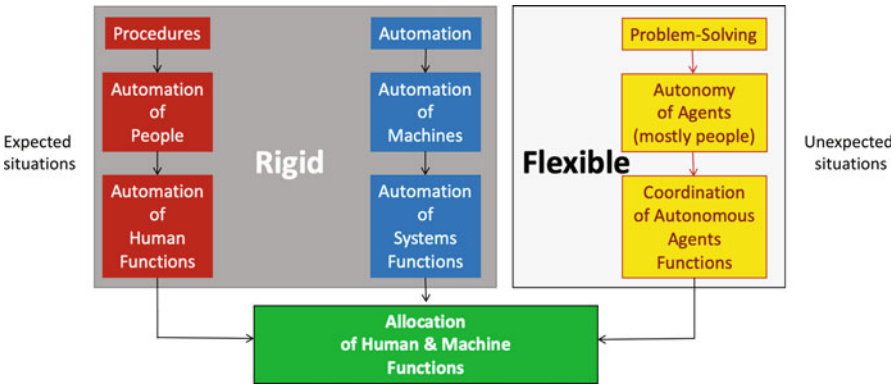


**Fig. 9** Procedures, automation, and problem-solving leading to the allocation of human and machine functions [6]

which involves deep learning. Case-based reasoning (CBR), as presented by Aamodt and Plaza [1], is very useful in the context of experience feedback management. CBR is based on four main processes: retrieval of one (or several) similar case (s) similar to a current case; reuse of previously used cases to elaborate a working-in-progress solution; revision of the working-in-progress solution by modifying its structures and functions until a satisfactory solution is found; recording of the successful solution for later reuse; and potentially making it more generic. CBR could be developed within a statistical framework in order to perform probabilistic inference as opposed to deterministic inference [64].

Machine learning (ML) has become dominant in AI because it offers algorithms that enable to reduce huge data sets to "meaningful" information. Consequently, ML can be very interesting within our flexible autonomy endeavor. Indeed, flexible autonomy should be based on experience, which is acquired incrementally through a large number of try-and-error activities and therefore big data sets. As a matter of fact, positive experience is as important as negative experience. If incidents and accidents are very well documented and can serve as useful data for learning, the focus should be put even more on positive experience (i.e., things that went well). ML algorithms are developed to make sense of large amounts of data. They enable the elicitation of patterns, information organization, anomalies and relationships detection, as well as projections making. These algorithms will enable fine-tuning of increasingly autonomous systems performance in a safe, efficient, and comfortable manner. They can contribute to improve task execution precision. Most important here is the definition of "meaningful" information, which cannot be done without well-done human-centered design. AI algorithms are not neutral; they incorporate human-made requirements and technological constraints that determine all logical and mathematical formula that they use.

Intelligent visualization [24, 33, 61] is a growing field of investigation that attempts to develop visualization methods and tools that enable complex data to be better understood by people. In other words, it helps people to be more familiar with complex systems when they are appropriately visualized. Remember the adage, "A picture is worth a thousand words." In addition to static pictures, dynamic animations, simulations, and movies can be displayed to improve understanding of complex data and concepts. AI can support intelligent visualization as demonstrated in visual analytics [35]. More specifically, human visual exploration could be supported by data mining for knowledge creation and management.

On the organizational side, three systemic interaction models can be considered [6]:

- Supervision is a process that enables a system (i.e., a supervisor) to supervise interactions among other systems that interact among each other. Supervision is about coordination. This interaction model is used when systems do not know each other and/or do not have enough resources to properly interact with each other toward a satisfactory performance of their constituting systems.
- Mediation is a process that enables systems to interact with each other through a mediation space made of a set of mediating systems, such as ambassadors and

diplomats. This 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 a socio-cognitive model of its environment to interact with the other systems, maximizing some kinds of performance metrics. Note that this model is collective and democratic. This 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. Other models could be used such as dominance of a system over the other systems (i.e., a dictatorial principle).

On the human side, people can be designers, engineers, developers, certifiers, maintainers, operators or end users, trainers and dismantlers (not an exhaustive list). People, in the TOP model, have activities and jobs. Anytime technology and/or organization changes, people may change their activities and/or jobs. Sometimes, new technology may lead to people losing their jobs, or conversely new jobs (i.e., functions) should be created and therefore a new set of people might be hired (i.e., a new structure should be created within the organization). People have their own human factors issues, such as fatigue, workload, physical and cognitive limitations, and creativity [6].

Sociotechnical SoS infrastructure can be hierarchical or heterarchical, for example. Evolution of digital organizations drastically changed people's jobs going from the hierarchical army model to the heterarchical orchestra model [9], with musicians, some of them being conductors and compositors. More formally, an orchestra playing a symphony (i.e., a product) requires five interconnected components:

- Music theory as the common language (i.e., a framework for collaborative work).
- Scores produced and coordinated by composers (i.e., coordinated tasks to be executed).
- Workflow coordinated by a conductor (i.e., system of systems activity).
- Musicians performing the actual symphony (i.e., the actual system of systems).
- Audience listening produced symphony (i.e., end users of the product).

## Concrete Chapter Contribution: The PRODEC Method

Human-centered design of complex systems is a matter of identification of the multiple human and machine entities, considered as systems, which can be physical and/or cognitive (cyber). Systems can indeed be modeled by roles, contexts of validity and resources that are systems themselves. Therefore, these properties need to be properly identified. PRODEC is a method to this end [12]. It is based on the distinction used in cognitive science and computer science between procedural and declarative knowledge. Procedural knowledge is about operations experience that is often expressed in the form of stories by subject matter experts. Declarative knowledge is about objects and agents required in the design of a

targeted human-machine system. The PRODEC method is articulated around the elicitation of procedural knowledge from subject matter experts and abduction of various human and machine systems properties and attributes. This abduction process is based on creativity and validation of systems being targeted. The PRODEC process may take several iterations to converge. It is highly recommended to run human-in-the-loop simulation for such validation and therefore incrementally create and maintain appropriate performance models and simulation capabilities.

## Procedural and Declarative Knowledge

The distinction between procedural and declarative (or logical) knowledge is not new. In the early stages of AI, this distinction was used to denote procedural and declarative programming. Procedural programming languages are high-level abstraction of computer instructions that enables the programmer to express an algorithm in a line-by-line sequence of instructions. Procedural programing languages originated from FORTRAN [34, 42] and include Pascal [66], C [51], and Python [21]. Conversely, declarative programming languages enable programmers to declare a set of objects that have properties and capabilities. They originate from LISP [4] and includes Haskell, Caml, and SQL, for example. Object-oriented programming originated from Smalltalk and was inherited from both paradigms, including Java and C++, for instance. These object-oriented languages enable programmers to declare objects and their properties as well as specific procedures called methods. Declared objects are further processed as they are by a software inference engine.

Computer science, AI, and cognitive psychology cross-fertilized for a long time. Indeed, procedural knowledge and its distinction with declarative (or conceptual) knowledge have been developed in several fields related to cognition such as educational science [41] and development psychology [56], including mathematics education [16, 27, 60], user modeling [20], and experimental psychology [38, 54, 63].

If the theater metaphor is used, a theatrical play is usually available first in a procedural manner. A writer produces an essay that tells a story. Then, a director selects, in a declarative manner, actors who have to read the essay and learn their roles and scripts procedurally and coordinate them. PRODEC has been designed to be used in HCD to benefit from both operations experience (i.e., human operators will be asked to provide their salient operations stories) and definition of objects and agents involved in the targeted human-machine system to be designed (i.e., the design team will provide prototypes at various progressive levels of maturity).

With this method, how operations are performed prior to starting any design is explored first. A procedural scenario is developed with experienced people, as a timeline of events. PRODEC is based on the claim that stories told by subject matter experts can be easily translated into procedural scenarios. Once several procedural scenarios are elicited, human-centered designers are able to extract meaningful objects and agents, which are described both functionally and structurally. This

constitutes declarative scenarios (i.e., organizational configurations). Of course, this articulation of procedural and declarative knowledge can, and should, be repeated as many times as necessary and possible to get a consistent and implementable human-machine system prototype, further developed and validated.

The production of procedural and declarative knowledge should be guided by a framework that supports the main factors that include artifacts to be designed and developed, users who will use them, the various tasks to be executed by these artifacts, the organizational environment where they will be deployed, and the various critical situations in which these tasks will be executed (see the AUTOS Pyramid in [10]).

## An Instance of PRODEC

At this point, let's provide an instance of the Business Process Modeling Notation (BPMN) (BPMN is based on a flowcharting technique tailored for creating graphical models of business process operations, similar to UML (Unified Modeling Language) activity diagrams. BPMN is procedural (i.e., it enables the description of procedural information with different graphical elements in the form of scripts, episodes, sequences, and so on, which mixes the ways agents interact with each other – it is a program or a routine in the computer science sense).) associated with an extended version of the Cognitive Function Analysis (CFA) method. BPMN is a standard for business procedural process modeling [48, 57] and a language that supports procedural knowledge elicitation and graphical formalization. CFA has been developed for declaratively identifying human and machine cognitive (and physical) functions and their relationships to support HCD [11]. CFA has been upgraded as Cognitive and Physical Structure/Function Analysis (CPSFA) to handle systems as agents [6].

Note that alternative procedural and declarative methods and tools could be used to instantiate the PRODEC method. The BPMN-CPSFA PRODEC method is then the following:

1. Elicit and review all tasks involved in the achievement of various goals.
2. Describe them as BPMN graphs (procedural scenarios).
3. Elicit cognitive (and physical) functions in the form of roles (associated to tasks and goals), contexts, and associated resources (declarative scenarios).
4. Describe and refine elicited resources' structures and functions (using CPSFA formalism).
5. Iterate until a satisfactory solution is found.

Resources are typically human and/or machine agents, in the AI terminology, and systems, in the SE sense. Contexts express persistent situations that can be either normal, abnormal, or emergency. Contexts are usually defined in the form of combined spatiotemporal conditions. For example, a postman job in a normal

context can be expressed in terms of time (i.e., every weekday from 8 am to 5 pm) and space (i.e., a well-defined neighborhood).

The PRODEC method has been used in an air combat system project called MOHICAN. This project aimed at deriving performance metrics to assess collaboration between pilots and cognitive systems, as well as trust in such cognitive systems. Before deriving such metrics, relevant human and machine functions should be elicited and further tested against such metrics.

First, task analyses were developed in the form of procedural scenarios (i.e., BPMN graphs). Then, function analyses were developed in the form of CPSFA declarative scenarios (agent-based configurations). Acquired air combat functions knowledge greatly determined the kinds of metrics that should be used for performance evaluation. For example, the "Acquire Information" function could be assessed from various viewpoints that include accuracy, time, workload, meaningfulness, and so on. This depends on context and available resources. Functions, either physical or cognitive, were declared in terms of role, context, and resources.

## An Illustrative Example of PRODEC Use

In the MOHICAN study, the BPMN-CPSFA PRODEC process describes (1) tasks to be completed within the cockpit; (2) their distribution among agents involved (e.g., Pilot and Weapon System Officer, decision-making assistant system); (3) required resources to complete each subtask (e.g., time, weapon system, air-to-air picture for situation awareness, etc.); and (4) the various agents as well as the interdependency between them (e.g., the pilot needs navigation information processed by the Weapon System Officer to achieve a subtask). When a satisfactory solution is found, it is typically implemented and tested into a human-in-the-loop simulation (HITLS). Testing results are then used to re-instantiate a BPMN-CPSFA PRODEC process. As an example, an emerging task, "Remind the pilot with safety altitude and safety heading," was discovered. This task is highlighted in the blue box in Fig. 10.

More specifically, when the situation degrades (e.g., autopilot height interception does not perform as expected), simulations have revealed the pilot's need for information (e.g., safety altitude reminder and heading to remain on the planned route) that experts didn't plan in their procedural projections. The main emergent function involved is "collaboration," which can be expressed in the form of role, context of validity, and resources required to make it useful and usable. It is clear in this example that the function "collaboration," implementing the task "remind the pilot with safety altitude and safety heading," can be allocated to either a human being (i.e., the Weapon System Officer) or a machine (i.e., a virtual assistant) thanks to an algorithm based on system status, flight parameters, and minimum height monitoring.

This example shows how emergent functions are discovered from an initial task analysis providing procedural scenarios (left hand side of Fig. 10), themselves used in HITLS (middle of Fig. 10) with subject matter experts (e.g., pilots), leading to

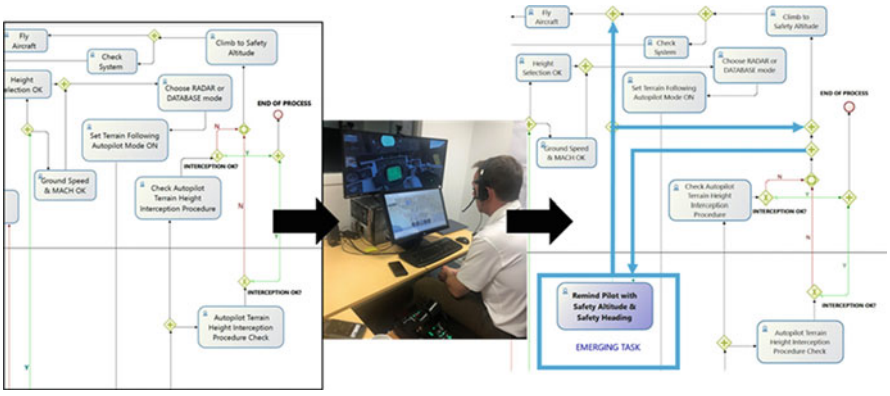


**Fig. 10** A MOHICAN project's example of PRODEC use

activity observation and further analysis and finally discovering emergent behaviors, properties, and functions (left hand side of Fig. 10).

## Discussion: Challenges, Gaps, and Possible Futures

### Departing from Technology-Centered MBSE

Even if systems engineering (SE) is based on a holistic approach, it is often too much technology centered and not enough human centered. Consequently, it fails sometimes spectacularly [14]. Uncertainty management is one of the main reasons. World changes very fast. Therefore, requirements and solutions should be constantly adapted. SE definitely requires flexibility. This is the reason why HSI should be better developed based on domain experience, creativity, HITLS, activity analysis, human-centered complexity analysis, as well as organization design and management [9]. Unifying HCD and SE will shape appropriate HSI and facilitate the production of successful systems.

The recent SERC research roadmap listed seven requirements relevant to AI4SE [43]: tools and domain taxonomies and ontologies; semantic rules; inter-enterprise data integration; automated decision framework; authoritative data identification; digital assistance; and digital twin automation. This roadmap specifies that HSI "will no longer be a specialty area but will be front and center to system definition." This chapter brings solutions for the operationalization of this endeavor. In addition, AI is considered as data science, including machine learning. Case-based reasoning, multi-agent systems, and human-robot interaction appear to be useful AI approaches and methods. What is called "hybrid Human/AI systems" is based on automated reasoning machine agents helping humans understand complex data. The term "virtual assistant" was used in the MOHICAN project. Human-autonomy teaming between humans and increasingly autonomous machines is a crucial endeavor.

## Human-Centered Modeling Limitations and Perspectives

What kinds of models will be useful and usable for HSI? Many human models for system analysis, design, and evaluation have been developed over the years. Let's cite a few. Originally, Simon and Newel's model [59] of information processing has been developed and extensively used in HFE and HCI. It was followed by Rasmussen's model [52] that provided a valuable and very much used framework for cognitive engineering. Besides these fundamental models, other more applied and targeted models were developed in the aerospace domain as support for simulation purposes, such as MIDAS [18] and MESSAGE [15]. These models attempted to mimic human behavior and cognitive processes (e.g., they were able to land an aircraft safely and mimic human errors). For example, MESSAGE has been used to figure out workload assessments during the certification of two-crewmen commercial aircraft. On another example, MIDAS supported exploration of computational representations of human-machine performance to aid designers of interactive complex systems by identifying and modeling human-automation interactions with flexible representations of human-machine functions. MIDAS helped producing guidelines in aeronautics and air traffic management [28].

Designers can work with computational representations of the human and machine performance, rather than relying solely on expensive hardware simulators and real flights with humans in the loop, to discover problems and ask "what-if" questions about the projected mission, equipment, or environment. The advantages of this approach are reduced development time and costs and early identification of human performance limits, plus support for training system requirements and development. This is achieved by providing designers accurate information early in the design process, so impact and cost of changes are minimal. After almost 40 years of experience of such simulated human models, it is obvious that there are several directions of investigation, such as virtual human-in-the-loop simulation (HITLS) where real humans interact with simulated machines and simple human models that support the development of metrics and scenarios useful for HITLS.

## HCD Based on Virtual Environments as Digital Twins

The shift from twentieth-century engineering associated with corrective ergonomics to twenty-first-century human-centered design based on digital prototyping, HITLS, and tangibility assessment opens SE to a radical transformation where HSI becomes central. Engineering design enables now to involve humans in the loop within a control and management virtual space, which incrementally becomes more tangible [8]. Figure 11 presents the tangibilization process of virtual HCD.

The term "control & management space" in Fig. 11 is generic, referring to a control room or a vehicle cockpit. Since it is deliberately assumed that the environment is multi-agent, agents being people and/or machines, initial agents being designed are virtual. These agents do not include the real people who are interacting with the control and management space within which agents are incrementally

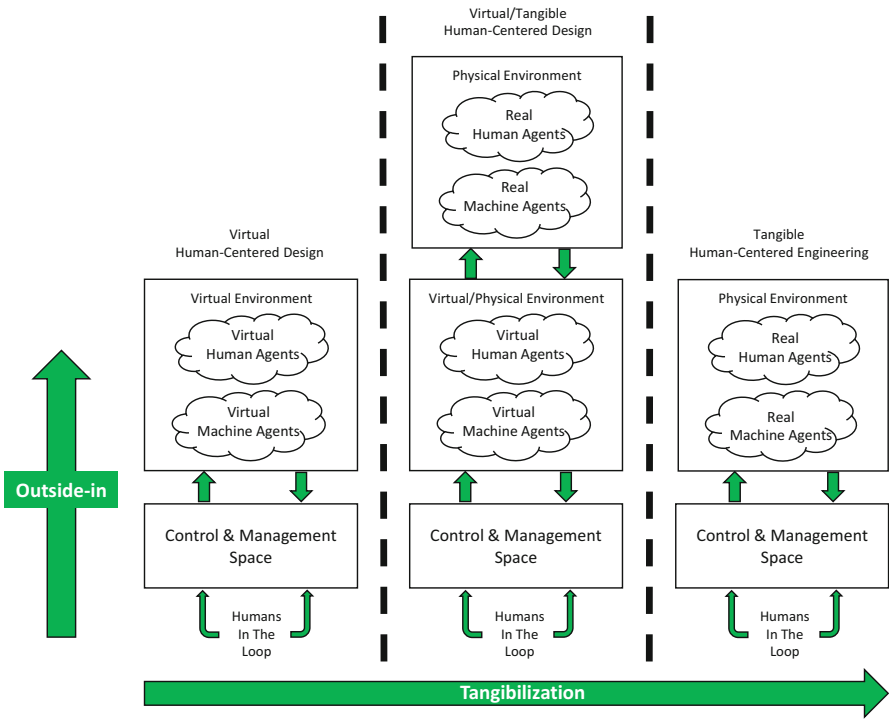


**Fig. 11** Tangibilization process in three steps: from virtual to tangible

tangibilized 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. A control and management room (space) is developed in the first place where real people will have to deal with a simulator of a virtual fleet of robots both moving and interacting with a virtual oil and gas offshore platform. Activities of these people are observed and analyzed to produce modifications of structures and functions involved in the simulation. This virtual HCD (VHCD) process is further pursued until a satisfactory design is reached, in terms of safety, efficiency, and comfort, for example. Then, one or two robots, as well as the platform, can be started to be tangibilized in a physical playground. A mixed 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.

In addition, once a human-machine system is fully developed, virtual environments that were used in VHCD can be used and refined as digital twins constantly evolving using experience feedback. In other words, system's interactive documentation is now available in the form of digital twins.

## Summary

Summing up, model-based human systems integration (MBHSI) is a very promising field of investigation. Possible directions of research are not only on human modeling but also on HITLS and more specifically human-centered digital twins, with tangibility in mind. This chapter presented an approach that enables improvement of design flexibility, resource management, and system knowledge though HITLS. Therefore, virtual prototyping that supports virtual human-centered design needs to be continuously improved.

New approaches are currently developed using HCD in MBSE [36] and more specifically virtual HCD in increasingly autonomous complex systems [3]. From a more general standpoint, using human and social sciences in systems engineering (i.e., HSI) differs from using hard sciences that include science, technology, engineering, and mathematics (i.e., STEM disciplines) by the fact that they force to explore human systems activity. More specifically, the main difference between conventional MBSE [23, 39, 53], that is, typically technology centered, and MBHSI relies on constant search for emerging behaviors through activity observation and analysis and, consequently, agile incorporation of emergent functions and structures into human-centered systems engineering practice.

An epistemological endeavor is in front of us. HSI requires more fundamental research efforts on human-centered and organizational topologies and ontologies that should support MBHSI. A novel system science where technology, organizations, and people (i.e., The TOP model) can be studied together in a rational and consistent manner needs to be harmonized.

## References


1. Aamodt A, Plaza E (1994) Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *Artificial Intelligence Communications,* 7(1):39–52
2. Amalberti R (2001) The paradoxes of almost totally safe transportation systems, *Safety Science*, 37:109–126
3. Bellet T, Deniel J, Bornard JC, Richard B (2019) Driver Modeling and Simulation to support the Virtual Human Centered Design of future Driving Aids. *Proceedings of INCOSE International Conference on Human Systems Integration (HSI2019),* Biarritz, France
4. Berkeley EC, Bobrow DG (eds) (1964) *The programming language LISP: Its operation and applications*. MIT Press, Cambridge, MA, USA
5. Bertalanffy L (1968) *General System Theory: Foundations, Development, Applications*, Revised ed., New York, NY, USA: Braziller
6. Boy GA (2020) *Human Systems Integration: From Virtual to Tangible*. CRC Press – Taylor and Francis Group, USA
7. Boy GA (2019) Cross-Fertilization of Human-Systems Integration and Artificial Intelligence: Looking for Systemic Flexibility. *AI4SE: Artificial Intelligence for Systems Engineering*. REUSE, Madrid, Spain
8. Boy GA (2016) *Tangible Interactive Systems*. Springer, UK
9. Boy GA (2013) *Orchestrating Human-Centered Design*. Springer, UK
10. Boy GA (ed) (2011) *Handbook of Human-Machine Interaction: A Human-Centered Design Approach*. Ashgate/CRC Press–Taylor and Francis Group, USA

11. Boy GA (1998) *Cognitive Function Analysis*. Praeger/Ablex, CT, USA
12. Boy GA, Dezemery J, Hein A, Lu Cong Sang R, Masson D, Morel C, Villeneuve E (2020) PRODEC: Combining Procedural and Declarative Knowledge for Human-Centered Design. FlexTech Technical Report, CentraleSupélec and ESTIA, France
13. Boy GA, Schmitt KA (2012) Design for Safety: A cognitive engineering approach to the control and management of nuclear power plants. *Annals of Nuclear Energy.* Elsevier. https://doi.org/10.1016/j.anucene.2012.08.027
14. Boy GA, Narkevicius J (2013) Unifying Human Centered Design and Systems Engineering for Human Systems Integration. In: Aiguier M, Boulanger F, Krob D, Marchal C (eds), *Complex Systems Design and Management,* Springer, U.K. 2014. ISBN-13: 978-3-319-02811-8
15. Boy GA, Tessier C (1985) Cockpit Analysis and Assessment by the MESSAGE Methodology. *Proceedings of the 2nd IFAC/IFIP/IFORS/IEA Conference on Analysis, Design and Evaluation of Man-Machine Systems*, Villa-Ponti, Italy, September 10–12. Pergamon Press, Oxford: 73–79
16. Carpenter TP (1986) Conceptual knowledge as a foundation for procedural knowledge. In: Hiebert J (ed), *Conceptual and procedural knowledge: The case of mathematics*. Lawrence Erlbaum Associates: 113–132
17. Caruso P, Dumbacher D, Grieves M (2010) Product Lifecycle Management and the Quest for Sustainable Space Explorations. *AIAA SPACE 2010 Conference and Exposition*. Anaheim, CA
18. Corker KM, Smith BR (1993) An architecture and model for cognitive engineering simulation analysis: Application to advanced aviation automation. *AIAA Computing in Aerospace 9 Conference*. San Diego, CA
19. Comte A (1865) *A General View of Positivism*. Translated by Bridges, J.H., Trubner and Co., 1865 (reissued by Cambridge University Press, 2009; ISBN 978-1-108-00064-2
20. Corbett AT, Anderson JR (1994) Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, *4*(4):253-278
21. Deitel PJ, Deitel H (2019) *Python for Programmers: with Big Data and Artificial Intelligence Case Studies*. Pearson Higher Ed, ISBN-13: 978-0135224335
22. Endsley MR, Jones DG (2011) *Designing for Situation Awareness: An Approach to User-Centered Design*. CRC Press, 2nd edn. ISBN 978-1420063554
23. Estefan JA (2008) *Survey of Model-Based Systems Engineering (MBSE) Methodologies*. INCOSE MBSE Initiative. Rev. B. International Council on Systems Engineering. San Diego, CA
24. Garcia Belmonte N (2016) Engineering Intelligence Through Data Visualization at Uber. (retrieved on July 29, 2019: https://eng.uber.com/data-viz-intel/)
25. Glaessgen EH, Stargel D (2012) The digital twin paradigm for future NASA and US Air Force vehicles. *AAIA 53rd Structures, Structural Dynamics, and Materials Conference*. Honolulu, Hawaii
26. Grieves M (2016) *Origins of the Digital Twin*. Concept Working Paper. August. Florida Institute of Technology / NASA. https://doi.org/10.13140/RG.2.2.26367.61609
27. Hiebert J, Lefevre P (1986) Conceptual and Procedural Knowledge in Mathematics: An Introductory Analysis. In: Hiebert J (ed), *Conceptual and Procedural Knowledge: The Case of Mathematics*, 2: 1–27. Erlbaum, Hillsdale, NJ, USA
28. Hooey BL, Gore BF, Mahlstedt EA, Foyle DC (2013) *Evaluating NextGen Closely Spaced Parallel Operations concepts with validated human performance models: Flight deck guidelines*. NASA TM-2013-216506. Moffett Field, CA: NASA Ames Research Center
29. Husserl E (1989) *Ideas pertaining to a Pure Phenomenology and to a Phenomenological Philosophy.* Second Book. Trans. R. Rojcewicz and A, Schuwer. Dordrecht and Boston: Kluwer Academic Publishers. From the German original unpublished manuscript of 1912, revised 1915, 1928. Known as Ideas II
30. Husserl E (1963) Ideas: A General Introduction to Pure Phenomenology. Trans. W. R. Boyce Gibson. Collier Books, New York. From the German original of 1913, originally titled Ideas pertaining to a Pure Phenomenology and to a Phenomenological Philosophy, First Book. Newly

translated with the full title by Fred Kersten. Dordrecht and Kluwer Academic Publishers, Boston, 1983. Known as Ideas I
31. International Council on Systems Engineering (INCOSE), A World in Motion - Systems Engineering Vision 2025, July 2014
32. Ionesco E (1996) *Between life and dream (Entre la vie et le rêve).* Interview with C. Bonnefoy. Collection Blanche, Gallimard, Paris, France
33. Kruchten N (2018) Data visualization for artificial intelligence, and vice versa. (retrieved on July 30, 2019: https://medium.com/@plotlygraphs/data-visualization-for-artificial-intelligence-and-vice-versa-a38869065d88)
34. Kupferschmid M (2002) *Classical Fortran: Programming for Engineering and Scientific Applications*. CRC Press. ISBN 978-0-8247-0802-3
35. Keim D, Andrienko G, Fekete JD, Carsten Görg C, Kohlhammer J, Melançon G (2008) Visual Analytics: Definition, Process and Challenges. In: Kerren A, Stasko JT, Fekete JD, North C., *Information Visualization - Human-Centered Issues and Perspectives*, Springer: 154–175
36. Kim SY, Wagner D, Jimenez A (2019) Challenges in Applying Model-Based Systems Engineering: Human-Centered Design Perspective. *Proceedings of INCOSE International Conference on Human Systems Integration (HSI2019),* Biarritz, France
37. Laurel B (1991) *Computers as Theatre*. Addison-Wesley. ISBN 0201550601.
38. Lewicki P, Czyzewska M, Hoffman H (1987) Unconscious acquisition of complex procedural knowledge. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *13*(4), 523
39. Long D, Scott Z (2011) *A primer for Model-Based Systems Engineering*. Vitech Corporation. ISBN 978-1-105-58810-5
40. Madni AM, Madni CC, Lucero SD (2019) Leveraging Digital Twin Technology in Model-Based Systems Engineering. *Systems*. https://doi.org/10.3390/systems7010007
41. McCormick R (1997) Conceptual and procedural knowledge. *International journal of technology and design education*, *7*(1–2):141-159
42. McCracken DD (1961) *A Guide to FORTRAN Programming*. Wiley, New York. LCCN 61016618
43. McDermott T, DeLaurentis D, Beling P, Blackburn M, Bone M (2020) AI4SE and SE4AI: A Research Roadmap. *InSight Special Feature*. Wiley Online Library. https://doi.org/10.1002/inst.12278
44. Minsky M (1986) *The Society of Mind*. Touchstone Book. Published by Simon and Schuster, New York
45. Muller MJ (2009) Participatory Design: The Third Space in HCI. *The Human-Computer Interaction Handbook*, Hillsdale: L. Erlbaum Assoc: 1061–1081
46. Nielsen J (1993) *Usability Engineering*. Academic Press, Boston, MA. Available on amazon.com
47. Piascik R, Vickers J, Lowry D, Scotti S, Stewart J, Calomino A (2010) *Technology Area 12: Materials, Structures, Mechanical Systems, and Manufacturing Road Map*. NASA Office of Chief Technologist
48. White, S.A. (2004). Business Process Modeling Notation. Retrieved on November 23: https://web.archive.org/web/20130818123649/http://www.omg.org/bpmn/Documents/BPMN_V1-0_May_3_2004.pdf
49. Piaget J (1952) The Origins of Intelligence in Children. New York: Norton
50. Piaget J (1954) *The Construction of Reality in the Child*. New York: Ballantine
51. Prinz P, Crawford T (2015) *C in a Nutshell: The Definitive Reference 2nd Edition*. Kindle Edition. O'Reilly Media, ASIN: B0197CH96O
52. Rasmussen J (1983) Skills, rules, knowledge; signals, signs and symbols, and other distinctions in human performance models. *IEEE Transactions on Systems, Man and Cybernetics.* Vol. 13: 257-266
53. Reichwein A, Paredis C (2011) Overview of architecture frameworks and modeling languages for model-based systems engineering. In *ASME Proceedings*: 1–9

54. Richard JF (1983) *Engineering logic versus use logic* (*Logique de fonctionnement et logique d'utilisation)*. Research Report, # 202, INRIA, Le Chesnay, France
55. Russel S, Norvig P (2010) *Artificial Intelligence – A Modern Approach*. Third Edition. Prentice Hall, Boston, USA. ISBN 978-0-13-604259-4
56. Schneider M, Rittle-Johnson B, Star JR (2011) Relations among conceptual knowledge, procedural knowledge, and procedural flexibility in two samples differing in prior knowledge. *Developmental Psychology*, *47*(6), 1525
57. White, S.A. & Bock, C. (2011). *BPMN 2.0 Handbook Second Edition: Methods, Concepts, Case Studies and Standards in Business Process Management Notation*. Future Strategies Inc. ISBN 978-0-9849764-0-9.
58. SEBoK (2020) Guide to the Systems Engineering Book of Knowledge (retrieved on 5 May 2020: https://www.sebokwiki.org/wiki/What_is_a_System%3F)
59. Simon HA, Newell A (1971) Human problem solving: The state of the theory in 1970. *American Psychologist, 26*(2), pp. 145-159. https://doi.org/10.1037/h0030806
60. Star JR (2005) Reconceptualizing procedural knowledge. *Journal for research in mathematics education*: 404–411
61. St. Amant R, Healey CG, Riedl M, Kocherlakota S, Pegram DA, Torhola M (2001) Intelligent visualization in a planning simulation. Proceedings Intelligent User Interfaces IUI'01). Santa Fe, New Mexico. ACM. 1-58113-325-1/01/0001
62. Tuegel EJ, Ingraffea AR, Eason TG, Spottswood SM (2011) Reengineering Aircraft Structural Life Prediction Using a Digital Twin. *International Journal of Aerospace Engineering*
63. Willingham DB, Nissen MJ, Bullemer P (1989) On the development of procedural knowledge. *Journal of experimental psychology: learning, memory, and cognition*, *15*(6), 1047
64. Wilson RA, Keil FC (eds) (2001) *The MIT encyclopedia of the cognitive sciences*. MIT Press, Cambridge, MA. USA
65. Winograd T, Flores F (1986) *Understanding Computers and Cognition: a new foundation for design*. Initially published by Ablex Publishing Corporation, Norwood, NJ. Now Addison-Wesley, Boston, MA, USA. ISBN 978-0-201-11297-9
66. Wirth N (1971) *The Programming Language Pascal*. Acta Informatica, Volume 1:35–63

**Guy André Boy,** Ph.D., is FlexTech Chair Institute Professor at *Paris Saclay University* (CentraleSupélec) and Chair of *ESTIA* Science Board, Fellow of the *Air and Space Academy*, and Chair of the *Human-Systems Integration Working Group* of *International Council on Systems Engineering* (INCOSE). He was University Professor and Dean, *School of Human-Centered Design, Innovation and Art*, and HCD Ph.D. and Master's Programs at the *Florida Institute of Technology* (2009–2017), as well as a Senior Research Scientist at the *Florida Institute for Human and Machine Cognition* (IHMC). He was Chief Scientist for Human-Centered Design at *NASA Kennedy Space Center* (2010–2016). He was member of the Scientific Committee of the SESAR program (*Single European Sky for Air Traffic Management Research*) from 2013 to 2016. He was the Chair of the 2012 *International Space University* (ISU) *Space Studies Program* (SSP12) FIT/NASA-KSC local organizing committee. He was Adjunct Professor at the *École Polytechnique* in Paris. He was Board Member of the Complex Systems Engineering Master Pedagogic Committee at *Paris Saclay University.* He was the President and Chief Scientist of the *European Institute of Cognitive Sciences and Engineering* (EURISCO, a research institute of Airbus and Thales). He co-founded EURISCO in 1992 and managed it since its creation to its closing in 2008. Between 1980 and 1991, he worked in artificial intelligence and cognitive science for ONERA (the *French Aerospace Lab*) as a research scientist and *NASA Ames Research Center* as the *Advanced Interaction Media Group* Lead (1984–1991). Engineer and cognitive scientist, he received his masters and doctorate degrees from the French Aerospace Institute of Technology (ISAE-SUPAERO), his Professorship Habilitation (HDR) from *Pierre and Marie Curie Sorbonne University*, and his Full Professorship Qualifications in Computer Science and Psychology. Boy actively participated to the introduction of *cognitive engineering* in France and its development worldwide. He was the

co-founder in 2004 of the *French Cognitive Engineering School* (ENSC) in Bordeaux. He co-founded the HCI-Aero Conference series, in cooperation with the *Association for Computing Machinery* (ACM). He is the author of more than 200 articles and two textbooks, *Intelligent Assistant Systems* (Academic Press, USA, 1991) and *Cognitive Function Analysis* (Praeger, USA, 1998), and the editor of the *French Handbook of Cognitive Engineering* (Lavoisier, France, 2003) and the *Handbook of Human-Machine Interaction* (Ashgate, UK) in 2011. His most recent books are *Orchestrating Human-Centered Design* (Springer, UK, 2013), *Tangible Interactive Systems* (Springer, UK, 2016), *Human Systems Integration: From Virtual to Tangible* (CRC Taylor and Francis, USA, 2020), and Design for Flexibility: A Human Systems Integration Approach (Springer Nature, Switzerland, 2021). He is a senior member of the ACM (Executive Vice-Chair of ACM-SIGCHI from 1995 to 1999), Corresponding Member of the *International Academy of Astronautics*, and Chair of the Aerospace Technical Committee of *International Ergonomics Association* (IEA).