

FlexTech: from rigid to flexible human-systems integration

Guy Andre Boy

▶ **To cite this version:**

Guy Andre Boy. FlexTech: from rigid to flexible human-systems integration. Ron S. Kenett; Robert S. Swarz; Avigdor Zonnenshain. Systems Engineering in the Fourth Industrial Revolution: Big Data, Novel Technologies, and Modern Systems Engineering, Wiley, pp.465-481, 2019, 978-1-119-51389-6. 10.1002/9781119513957.ch18 . hal-02956588

HAL Id: hal-02956588
https://hal.science/hal-02956588

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.

Boy, G.A. (2019). FlexTech: from rigid to flexible human-systems integration Chapter 18, in Systems Engineering in the Fourth Industrial Revolution: Big Data, Novel Technologies, and Modern Systems Engineering, R.S. Kennett, R.S. Swarz & A. Zonnenshain (Eds.), ISBN: 978-1-119-51389-6 December.

Chapter 18

FlexTech: from rigid to flexible human-systems integration

Guy A. Boy

Synopsis

Is Human-Systems Integration (HSI) a necessary component of Systems Engineering (SE) or the opposite (i.e., SE a necessary component of HSI)? It all depends on the perspective! If you are a technology-centered engineer, SE will provide you with methods and tools to develop technological systems, and you will need human factors specialists to develop user interfaces and test usability of the end product. In this perspective, HSI is a necessary component of SE. However, if you are human-centered designer, you will need methods and tools to design and develop systems that integrate human and machine requirements from the very beginning of design to decommissioning of the system of systems at stake. This states the question of what we mean by "system." A system is simply a representation that helps figuring out physical and cognitive functions and structures of both people and machines. In this chapter, several areas related to HSI are covered, including task and activity analysis, cognitive engineering, organization design and management, function allocation, complexity analysis, modeling and human-in-the-loop simulation (HITLS). Contemporary HSI design approaches are supported by virtual HITLS, which involves tangibility issues. A discussion is started on the various kinds of data that should be collected and tangibility indicators to develop appropriate HSI. An aeronautical example is provided to illustrate how HSI should be developed in the design and development of a system of systems. We conclude on the necessary shift from rigid automation to flexible autonomy that prefigures the *FlexTech* (i.e., technological solutions, associated with organizational setups and human functions, that improve operations' flexibility).

Keywords: human-systems integration, human-centered design, FlexTech, system science, systems engineering, systems of systems, creativity, innovation, experience, human-in-the-loop simulation, agile development, multi-agent modeling, TOP model.

18.1 Industry 4.0 and Human-Systems Integration

The 4th industrial revolution, that is also called Industry 4.0, results from the growing digitalization of industrial organizations, and more generally our sociotechnical society. Depending on approach and background, several new trends are popping up and developing: automation is leading toward cyber-physical systems; computer science to the Internet of things; computer engineering to cloud computing; and human-computer interaction and artificial intelligence to cognitive computing. The 3rd industrial revolution was technology-centered, considering human-systems integration as an adaptation of people to machines being developed. After World War 2, Human Factors and Ergonomics (HFE) developed as a discipline attempting to make machines usable by people, either anybody (public use) or experts (mostly in life-critical systems). HFE was initially focused on physical ergonomics, as well as health and safety at work. HFE was handled by physicians until the beginning of the 1980s when microcomputers invaded our societies. Cognitive ergonomics started

then to be prominent. At the same time, human-computer interaction became a requirement to enable people to commonly use computers in their everyday lives. As a matter of fact, Human-Computer Interaction (HCI) started to be developed as a discipline, involving computer scientists and cognitive scientists. HCI led to cognitive engineering, graphical user interfaces (GUIs) and computer-supported cooperative work (CSCW), for example. At the same time, systems engineering developed to structure industrial organizations with technical processes. Several concepts started to be studied and operationalized such as systems of systems, agile development (Schwaber, 1997, 2004), and model-based systems engineering. However, these concepts and approaches were mostly technology-centered and required to be revisited to include humans and organizations at the center. Human-Centered Design (HCD) then developed as a discipline. From the beginning of the 21st century, it became possible to effectively include human and organizational requirements into design and development processes. Why? Simply because human-in-the-loop simulations (HITLS) could be taken seriously at design time, enable observing people's activity, virtually design and test possible solutions, and derive appropriate human and organizational requirements. Human-Systems Integration (HSI), which integrates HCD and systems engineering during the whole life cycle of a system (including people and machines), became a discipline.

Consequently, virtual prototyping has become a precious method supported by digital modeling and simulation tools that supports HSI. People can be included from the beginning of the design process to the end of the life cycle of a system. From the beginning of the 21st century, we have inverted the engineering approach. Instead of going from hardware to software (i.e., constructing structures and then refining functions), we now go from software to hardware (i.e., designing and testing functions and then deducing structures) (Boy, 2017). Nowadays, we can 3D print structures from digital models, which have been functionally tested (in virtual environments). Products can be certified virtually before they are physically developed. This is very appealing, but the emerging concept of tangibility should be better mastered and tested. We will further describe and analyze tangibility in this chapter. Each system (e.g., a modular equipment) can be considered as a Tangible Interactive System (TIS) that can be easily manipulated, both structurally and functionally (Boy, 2016). This TIS modeling and approach are very useful for managing rapid changes. Indeed, visualization of explicit knowledge of interconnected manufacturing systems, as virtual twins, provide useful and usable support for reconfiguration.

HSI will then be understood using a systemic approach. More specifically, we will specifically focus on real-time process control, data interoperability, consistency and integration based on a System of Systems (SoS) approach. SoS framework is very useful for mastering systems' structures and functions that lead to reducing risks, controlling costs, and improving function allocation. In other words, considering human factors from the beginning of design and all along the life cycle of systems improves safety, efficiency and comfort. The SoS framework offers mediating concepts and tools that shareholders can use for collaborative work (offering a common frame of reference), distributed decision-making and shared situation awareness.

The orchestration (Boy, 2013) of various kinds of TISs requires developing an ontology of the manufacturing domain being investigated (i.e., music theory of we use the Orchestra metaphor), making prescribed tasks explicit (i.e., analog to scores provided to the musicians that can be humans or machines), coordinating these tasks at design time (i.e., analog to the role of the composer), coordinating produced activities at operations time (i.e., analog to the role of the conductor), training the various TISs and human operators for improved performance and cooperation (i.e., analog to musicians learning), and seriously considering end-users of products being manufactured (i.e., analog to the audience of a symphony). This approach will provide flexibility during the whole life cycle of systems being developed. We call resulting technology, FlexTech.

HSI applies to all stakeholders involved during the life cycle of a system. We should not restrict HSI to end users. Designers are main actors of HSI. Think about designers of kitchen appliances that should be adapted to customers' needs and requirements. There are two extreme strategies: (1) procedural manufacturing of very well dichotomized generic systems; and (2) expert-based crafting of customized systems. In the same way, fashion designers use the distinction: "ready to wear" versus "tailored." If we want to increase flexibility of production for customized products, we need to develop flexible technology and knowledgeable/skilled employees. Designing such FlexTech requires to investigate and use approaches such as systems of systems, teams of teams, function allocation, failure management and creativity. In addition, we should not only develop systems in "normal situations", but also consider "abnormal and emergency situations" that may leads to "unexpected situations" where people have difficulty and even impossibility to find appropriate solutions in a reasonable amount of time. This is the reason why we propose to carry out an HSI study that considers tangibility of digital systems being developed in terms of complexity, stability, flexibility, maturity and sustainability of the system being developed and its usages (Boy, 2016).

How should we proceed? For example, we could define and develop several use cases, which could be derived from expert experience, accident and incident analyses and various possible projections based on current societal evolution. Two contradictory concurrent approaches could be mixed: (1) creativity and innovation (divergent processes); and (2) use of experience (convergent process). Such use cases usually lead to the development of scenarios in terms of systems configurations and chronologies (i.e., storytelling, scripts like for a theater play). A prototype of the system is then developed and used to run HITLS in order to discover emerging patterns. The cognitive function analysis technique (Boy, 1998) could be used to model the resulting system of systems. The resulting model could then be used to support assessments and validation of the system of systems being developed. Principle and criteria will be developed for such assessments.

Summarizing, HSI requires us to develop virtual prototypes, scenarios and socio-technical criteria that will support our HCD approach. It is by concurrently developing prototypes that incrementally become more tangible (physically – hardware part – and cognitively – software part) that we will reach these objectives. Current understanding of what tangibility of digital technology means is developed in this chapter.

18.2 HSI evolution: from interface to interaction to organizational integration

It would be difficult to make sense of HSI without defining the distinction between task and activity. A task is what is prescribed to be done. An activity is what is effectively done once the task is executed. Task analysis has been extensively used to support the design of user interfaces. However, usability studies have shown that activity observation and subsequent analysis are crucial to improve interaction design (Kaptelinin & Nardi, 2006). For a long time, problem was that activity was only observable on existing systems whether currently used systems prior to novel design or once a new system was fully developed. On the one hand, activity analysis on systems, which will become obsolete once a new system will be developed, forces continuity and not disruptive innovation. On the other hand, activity analysis once a new system is fully developed is likely to show design flaws that will or will not be possible to fix system-wise; only cosmetic patches can be brought at the user interface level. This has been done for a long time.

Historically speaking (Figure 1), HFE developed activity analysis. However, activity analysis was only possible using an existing system (i.e., before the design of a new system using the old system when it existed, and after it was fully developed and usable). HFE then led to corrective ergonomics.

In HCI, as a discipline, there is no, or very little, difference between task and activity in the manipulation of computers. This is the reason why task analysis was heavily used in HCI.

Interaction design was born in HCI. However, HCI has been very often limited to computing systems where complexity of multi-physical systems was almost absent. An aircraft, an air traffic control system or a nuclear power plant is such a complex system, where observed activity is generally different from related prescribed tasks, typically provided in the form of operations procedures. Related user interfaces, whether they are called cockpits or control rooms, involve deeper considerations than generic desktop graphical user interfaces.

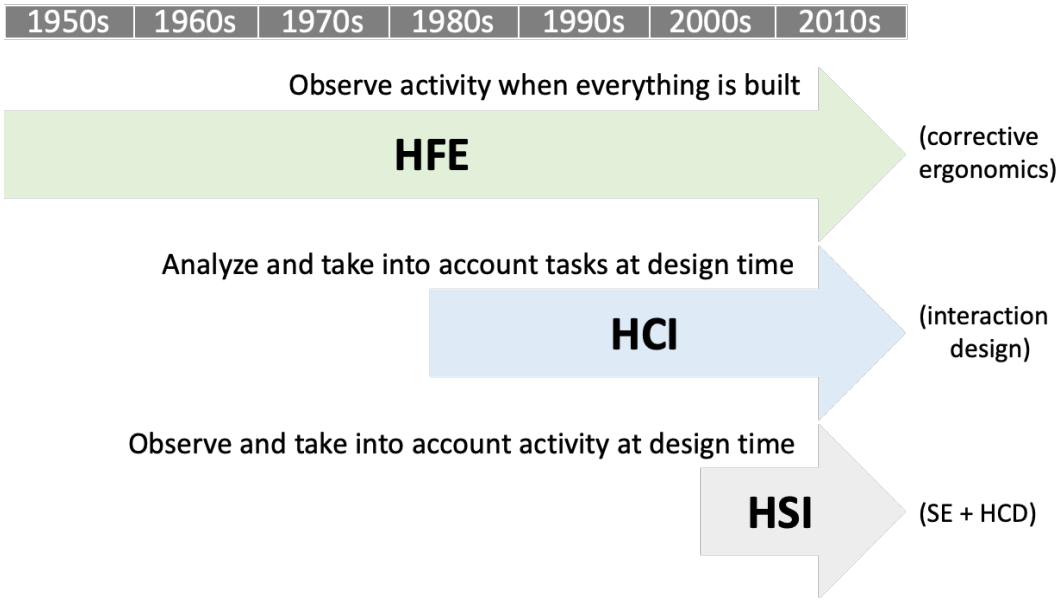


Figure 1. Evolution of human-centered technology approaches.

Since the beginning of the 21st century, systems engineering started to become aware of the importance of human factors in engineering design. More specifically, defining user interfaces when a complex system is developed is not satisfactory. Many user interfaces tend to hide engineering design flaws, and unfortunately adapt people to developed systems, creating situation awareness and decision-making problems in critical situations.

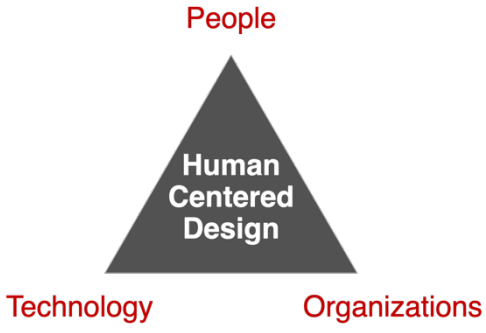


Figure 2. The TOP model for Human-Centered Design.

This is an important reason why HSI started to develop in engineering design. which leads to systems engineering where systems include people and organizations. It is common to hear about sociotechnical systems to denote such systems. HSI attempts to concurrently consider Technology, Organizations and People (TOP) during the whole life cycle of a system (Figure 2) and requires a more formal "system" definition.

18.3 What does the term "system" mean?

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

A system can be either cognitive (or conceptual), physical or both (Boy, 2017). It also has at least a structure and a function (Figure 3). Today, machines have cognitive functions (e.g., the cruise control function on a car enables the car to keep a set speed). Figure 3 presents a simple ontological definition of the "system" representation.

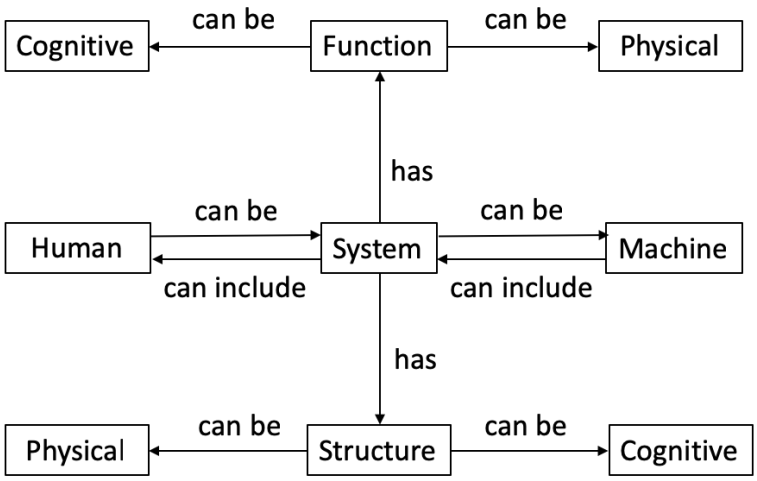


Figure 3. The cognitive-physical structure-function approach of the system representation.

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

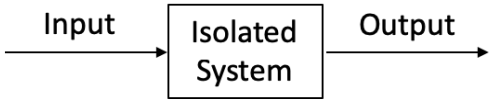


Figure 4. Single-agent isolated system.

As a matter of fact, an agent can be defined as a society of agents (i.e., an agency is an agent itself) in Minsky's sense (Minsky, 1985). For example, a postman is an agent of an agency that is commonly called "The Post". In this chapter, the term "system" will be synonym of "agent." It can be said that a system is a representation of an agent, whether it is a human or a machine. In the same way an agent is a society of agents, a system is a system of systems. Therefore, system's structures and functions can be defined as structures of structures and functions of functions. More generally, it is now common to use the system-of-systems concept to denote sociotechnical interconnected systems (Figure 5).

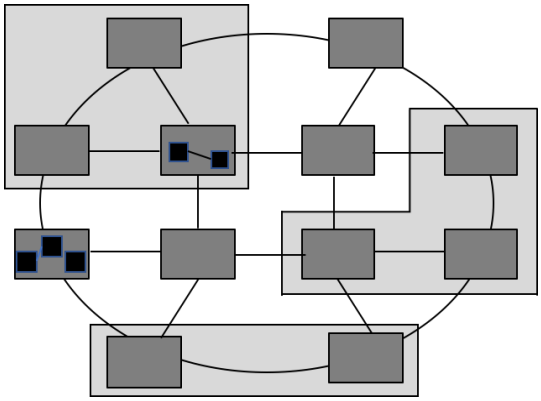


Figure 5. Interconnected system of systems.

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

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

Context of validity of postman's role (i.e., delivering letters) can be defined by a time context (e.g., from 8:00 am to noon and from 2:00 pm to 5:00 pm) and a space context (e.g., the neighborhood where he/she has to deliver letters). Context can be normal (i.e., every day the same) or abnormal (e.g., some other postmen are absent and he/she needs to expand his/her time and/or space context). Postman's resources can be physical (e.g., a bag and a bicycle) or cognitive (e.g., a pattern-matching cognitive function that enables him/her to put each letter in the right box). At this point, it becomes clear that a function is a function of functions (e.g., postman's function to deliver letters is a function of another function, the pattern-matching function).

More generically, function's resources are systems. They can be either physical, cognitive or both. In many cases, systems may be represented by their structures. This representation is very convenient for function allocation in a system of systems (i.e., among systems [or structures] in a network of systems).

In the same way, contexts can be represented as contexts of contexts. In aviation for example, the overall flight context can be decomposed into smaller contexts that include taxiing, takeoff, after-takeoff climb, cruise, descent, approach and landing, and so on. Each of these contexts can be decomposed into even smaller contexts. A system may evolve from Context 1 to Context 2 (Figure 6), where Context 1 may represent a normal situation and Context 2 an abnormal situation, for example.

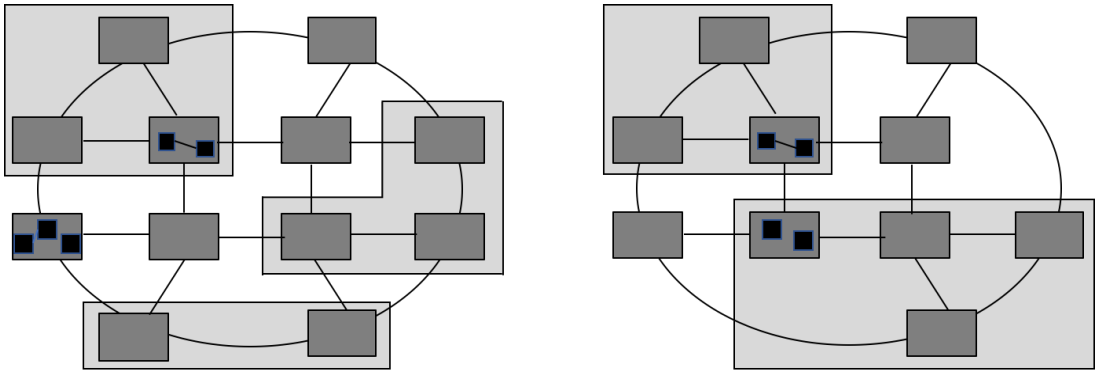


Figure 6. Evolution of system of systems from Context 1 to Context 2.

18.4 HSI as function allocation

Fitts and his colleagues were the first to be technically interested in function allocation among people and safety-critical machines (Fitts, 1951). In this chapter, function allocation will be brought a step further. Instead of allocating functions statically only using a task-based approach, the system-of-systems representation enables systems being incrementally designed by redefining their functions using an agile activity-based approach. That is to say that, since system-of-systems models can be simulated (using explicit system representation), emergent properties, and emerging functions, can be reallocated into the system of systems (DeLaurentis, 2005). Let's better explain how this approach works and can be effectively used. Here is a procedure:

1. A system of systems is defined from innovative concepts, provided by creative people, consolidated by experience, provided by domain experts. A first set of functions is defined and allocated to the various human and machine systems. Systems should be defined using the ontology provided in Figure 3.
2. The system of systems is simulated, producing an activity that should be observed and analyzed. Activity observation should be guided by appropriate metrics to be defined with respect to tangibility issues.
3. Effectivity of original functions should be analyzed and emerging functions induced from activity analysis. A new function allocation should be performed.
4. If there are still open questions on current function allocation, another sequence 1-2-3 should be executed, else the allocation process is finished.

It is clear that such an approach can be very sensitive to chosen levels of granularity of systems (structures) and functions. The finer gain, the longer and more difficult the process will be. Conversely, the bigger grain, the easier but less informative the process will be. As in all processes of that type, expertise and experience matter. This is the reason why one has to practice before becoming effective and efficient. It is also clear that such a function allocation process is never finished because the world in which the system of systems will work is open, and therefore unexpected events will always create new emergent properties that need to be discovered, analyzed and considered in a new allocation process. However, it is crucial to define principles and criteria that will help stop momentarily the allocation process to accept a "good-enough" solution validated by experienced people.

18.5 The tangibility issue in human-centered design

HITLS and therefore HCD are based on software-based models that can be considered as digital twins of the system being developed and incrementally optimized. Tangibility is inversely proportional to the distance between such digital twins, also called virtual prototypes in design and development phases, and real systems that they represent.

Tangibility can be defined from two complementary perspectives: physical and figurative (Boy, 2016). Physical tangibility denotes the capacity of an object or a system to be grasped, held and manipulated correctly. Figurative (or cognitive) tangibility denotes the capacity of an argument, an abstraction or a concept to be grasped, held or manipulated correctly. Tangibility is both related to realism and meaning. When something makes sense, it is tangible. It may make sense sensitively (physically) and/or cognitively (figuratively).

Tangibility should be assessed using appropriate properties and metrics. It can be decomposed into five considerations that lead to such metrics: complexity; maturity; flexibility, stability; and sustainability. Just a few properties and metrics will be provided in this chapter as illustrations. A deeper account will be available in a book that should be published soon (Boy, 2020).

Considering the complexity of a system seriously requires finding out what parts are separable, in the sense of being investigated separately without disturbing the overall structure and function of the system. For example, an organ of a biological system is separable when it can be studied in isolation of the overall system it belongs to. Some organs are simply not separable (e.g., the brain of a human being should be investigated and treated connected with the rest of the body; it is not separable). Consequently, complex systems have important metrics that cannot be measured through independent variables. For example, workload of a person cannot be measured by simple variables such as heart rate or electroencephalographic signals. Indeed, workload is not only an output that people produce, it is also an input that people need to regulate their performance. Workload is an emergent property. Therefore, workload deserves to be modeled in order to qualify as a human-factors metrics. For example, time-wise workload could be expressed as the ratio of required time on the available time to perform a task. Workload is one of the metrics that can be used to assess tangibility of a complex human-machine system.

In the same way, maturity should be continuously assessed in order to make sure that we correctly insure both physical and figurative tangibility when the system will be delivered. Maturity can be explored in three ways. Technological maturity expresses the amount of confidence one could have in using the system being developed without needing expert technical knowledge. For example, very few people considered computers as tangible during the 1970s because computing technology was not mature. Today, almost everybody tangibly uses a computer every day. Maturity of practice expresses the amount of virtuosity one could have in performing any task using the system. For example, there are various kinds of computing systems for specific usages such as smart phones usable for communication purposes and digital vision systems to detect intruders in specific venues. Organizational or societal maturity expresses the amount of lack of resistance an organization or a society has to effectively use the system. For example, our society was not ready (i.e., societally mature) in the beginning of the 1990s to recognize the Apple's Newton tablet as a tangible tool, even if it was a great system. It took about 15 years to make tablets (e.g., iPad) acceptable for our societal needs.

Flexibility is always needed when things go wrong. Designing a system that provide rigid alternatives in critical situations requires to be thought twice before committing. Indeed, people are

always more flexible than machines. This means that people can provide solutions that machines could never find if they are not programmed to do so. Conversely, even if people are unique creators of ideas, they may make mistakes. Consequently, in life-critical systems, flexibility should be bounded within reasonable limits, and people should be equipped with appropriate tools that support them in problem-solving tasks. Various kinds of safety solutions can be developed going from fail-safe systems to fault-tolerant systems.

Stability can be passive or active depending on the autonomy of the system. Passively stable systems are able to recover from various kinds of disturbances. Actively stable systems require external assistance to keep safe performance and sound activities. The more a system is autonomous, the more it can go back to a stable state when disturbed or faced with an unexpected event. Unstable systems are generally not considered as tangible without considering stabilization systems that go with them. Autonomy will be further defined later in this chapter.

Finally, sustainability is also related to autonomy, but in the sense of the relationship between the system being developed and its environment. A system is totally sustainable when it is self-sufficient in a natural context and does not destroy its environment. At this point, tangibility deals with philosophical models where a choice should be made between whether ecology leads economy or the opposite.

18.6 Automation as function transfer

Automating a machine is delegating a human function to a machine. Obviously, the resulting artificial function is not exactly doing what a human can do, but it is close enough to say that the machine owns a cognitive function that human used to have (Boy, 1998). For example, the function "heading control" in an aircraft cockpit was transferred to the machine in the early 1930s (e.g., Boeing 247), in the form of an autopilot. Autopilots have been used since then on aircraft. During the 1980s, commercial aviation introduced integrated and digital autopilot and auto-throttle that defined a new control loop, the guidance loop based on high level modes. This was a clear evolution. At the same time, the flight management function was introduced on commercial aircraft, which integrated guidance and flight management on a computer, called the flight management system (FMS). This was a drastic revolution, moving flying activities from control to management, that is from controlling basic flight variables to managing systems, which themselves are controlling basic flight variables. This was a delegation shift. From then, pilots had to collaborate with the FMS. However, these changes were very much between pilots and aircraft systems, where each pilot was considered as a single agent. The aviation community learned over the years that technology introduced another factor, that is the organization among involved agents.

Indeed, the TCAS (Traffic-alert and Collision Avoidance System), introduced during the 1980s, had a significant impact on the reduction of midair collisions (Kuchar & Drumm, 2007). It also introduced a new concept of authority sharing between air traffic controllers (ATCOs) and pilots. The question was no longer installing new systems onboard without taking care of other agents or systems outside the cockpit, but considering air and ground agents or systems. Technology, organizations and people (Figure 2) had to be "designed" concurrently. Using TCAS is not a trivial thing to do. The TCAS system has three internal functions: a surveillance function (and system); a trajectory extrapolation function (and system); and a threat detection function (and system). The surveillance function, F_S, monitors presence of other aircraft in the vicinity. The trajectory extrapolation function, F_E, calculates, with respect to range, bearing and altitude, the possibility of conflict with another aircraft if detected by F_S. The threat detection function decides if there is a

threat with respect to F_E result. However, technological functions of the TCAS are not sufficient to solve the entire problem of collision avoidance. Indeed, the pilot is also involved. Once a possible conflict is detected, TCAS provides a "Traffic Alert" to the pilot (aircraft's traffic alert function F_{TA}), in the form of two possible resolution advisories (aircraft's resolution advisory function F_{RA}: "Climb" or "Descent"). Then, the pilot needs to quickly decide (pilot's decision function F_D), inform the ATCO (pilot's response selection function F_{RS}), and maneuver accordingly (pilot's maneuver function F_M). Normally, the pilot should obey the TCAS and not the ATCO (in the case the ATCO asks the pilot to "Climb" or "Descent"). The 2002 Überlingen (Germany) disaster, which killed 71 people, is an unfortunate example of such confusion using TCAS (Figure 7).



Figure 7. Überlingen accident (picture from the German Investigation Report, 2004).

Orders coming from TCAS and ATCO were contradictory, and one of the pilots obeyed the ATCO. Consequently, both conflicting aircraft maneuvered a "Descent"… They collided into each other. An HSI study, using a cognitive function analysis (Boy, 1998, 2011), would have proposed that a technological link between TCAS and ground control should be implemented in order to provide ATCO with correct information of TCAS resolution advisory to pilots. In such a system of systems, coordination is key. Without an HSI approach, technology-induced human errors remain quite possible (e.g., Überlingen disaster).

This authority sharing trend is now developing as air traffic density is increasing (4.5% per year for the last 35 years). The growing number of aircraft in the sky, more specifically on top of airports, deserves new kinds of studies on air traffic management complexity. The way air traffic should be handled in high density zones is not the same as what controllers used to be doing when the number of aircraft was reasonable to control. SESAR and NextGen programs are developing possible solutions for this kind of problem. 4D trajectories for example are planning-based solutions that still deserve investigations since planning and flexibility are contradictory. Indeed, air traffic is planned, the more it can be stabilized and handled more effectively, but also the more planning rigidifies and does not provide enough flexibility when failures or unexpected events occur. Consequently, current airspace system has to be considered as a system of systems.

18.7 From rigid automation to flexible autonomy

When everything is "normal", well-done automation works perfectly without human assistance. However as already stated above, in abnormal and emergency situations, automation tends to introduce rigidity, due to the fact that it executes procedures that cannot be changed easily. In case of

machine failure, human error or, more generally, unexpected events, people require flexibility to handle corresponding situations. We have been focusing on the human error syndrome for a long time, considering that people were "the problem." It is time to consider that people can be "the solution," when human presence is necessary. This is the reason why not only HSI is essential, but also people's competence and skills are crucial. In other words, the question of autonomy needs to be considered on the human side first during the whole life cycle of a human-machine system.

At this point, it is crucial to better define what "autonomy" means. A human or machine system is autonomous when he/she is able to handle (almost) any situations without external help. An autonomous system is equipped with sensors that enable to provide appropriate information leading to appropriate decision-making and action. Situation awareness is a typical cognitive function that includes perception of the situation, comprehension and projection to anticipate and perform correct actions (Endsley, 1995ab, 1998; Endsley & Garland, 2000; Boy, 2015). There is still a long way to go for identifying appropriate variables that will be implemented in situation awareness support on increasingly-autonomous systems. For example, birds flying in flocks have super TCAS systems that zoologists typically model in the form of three types of functions that can be simulated using differential equations: separation, alignment and cohesion (Potts, 1984; Reynold, 1987; Ballerini et al., 2007; Pemmaraju, 2013). These functions could be very purposeful to be adapted and implemented on aircraft for handling self-separation for example in highly congested environments. Of course, it goes without question that such solutions should be heavily tested in HITLS to look for emerging properties and functions that need to be further considered. In any case, the more aircraft will become autonomous, in the sense of providing more autonomy to pilots and also to aircraft themselves, the more coordination rules will be necessary. Again, without HITLS experimentations, such rules will not be discovered.

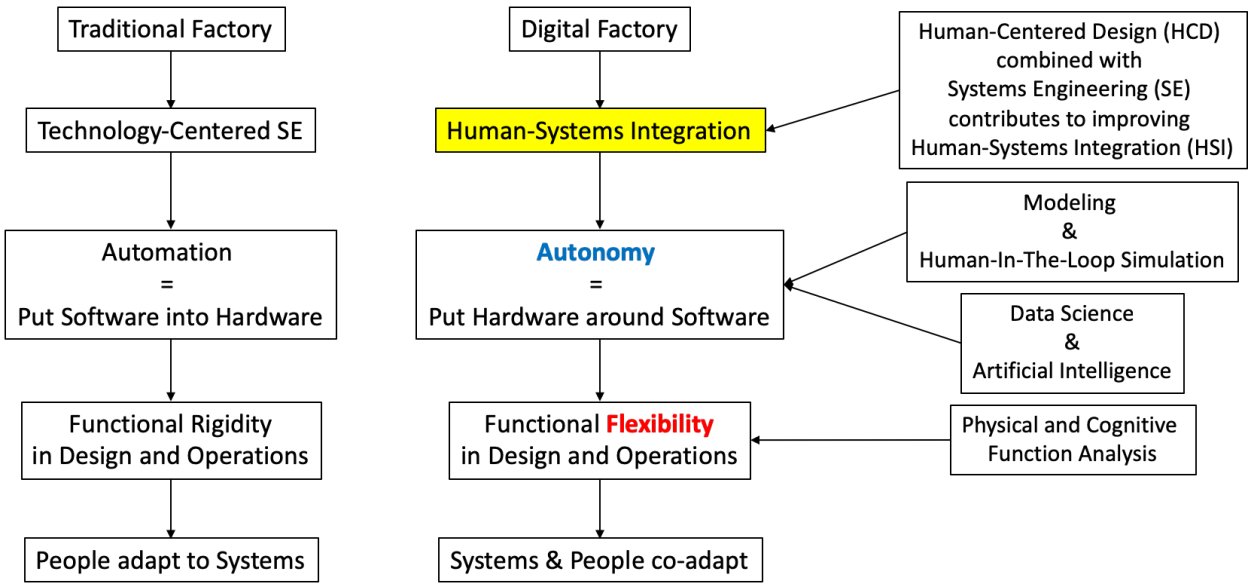


Figure 8. How could a digital world provide more autonomy and flexibility?

Let's summarize what the shift from rigid automation to flexibility means (Figure 8). The traditional factory typically led to technology-centered systems engineering, as current digital factory is progressively leading to human-systems integration (HSI) because human-centered design combine with systems engineering contributes to improve HSI. Automation contributed to put software into hardware, as autonomy should contribute to put hardware around software using modeling and HTILS. Traditional functional rigidity in engineering design and operations can nowadays leave the

floor to flexibility using HSI methods and tools, and more specifically physical and cognitive function analysis. Our evolving digital world provides more autonomy and flexibility where systems and people co-adapt instead of adapting people to systems. As already defined, systems include people and machines.

18.8 Conclusion

Human-systems integration is both a process and a product, which should take place as early as possible during the design process and evolve during the overall life cycle of the product. While human factors have been considered in the past by HFE and HCI specialists, being able to observe activity at design time is a brand-new capability, because modeling means are now available and realistic enough to support human-in-the-loop simulations.

Human-centered design combined with systems engineering approaches, such as agile development and systems-of-systems, effectively supports HSI. Indeed, technology, organizations and people must be considered concurrently during the whole life cycle of a system in which functions can be incrementally allocated to more appropriate agents with respect to principles and criteria. This means that HSI is no longer a matter of adapting people to machines by crafting user interfaces and operations procedures, but an integrated approach based on the concept of system of systems that integrates people and machines into articulated socio-technical systems. In fact, HSI theory is far from being fully developed, and requires more formal investigations and heuristic experimentations. More specifically, HSI currently leads to investigating autonomous complex systems that could be better designed, where autonomy should be thought for all agents, whether people or machines.

Finally, HSI developed using digital twins should be based on strong tangibility principles and assessment criteria, properly defined along five directions: complexity, maturity, flexibility, stability and sustainability. Number and content of these dimensions will be probably extended with respect to ongoing research and innovation developments. Instead of 20th century's automation (i.e., incorporation of software into hardware) leading to rigid operations, especially in critical situations, we promote a 21st century's approach based on coordinated autonomy of technology, organization and people (i.e., deducing appropriate hardware structures from software functionalities) leading to more flexible operations. This is the birth of the FlexTech.

18.9 Summary

Human-systems integration (HSI) denotes an evolution of conventional Human Factors and Ergonomics (HFE) discipline that focuses on evaluation of existing systems and usages, as well as Human-Computer Interaction (HCI) that provides methods and tools for interaction design. HSI evolution consists in considering the TOP model that supports symbiotic integration of Technology, Organizations and People from the beginning of design to the end of the life cycle of a system. Resulting technology is called FlexTech, which enables more flexibility in systems operations.

HSI can be seen as the association of Human-Centered Design (HCD) and Technology-Centered Systems Engineering (TCSE) where multi-agent modeling and Human-In-The-Loop Simulation (HITLS) are used from the beginning of the design process to increase knowledge included in the corresponding TOP model, keep enough flexibility during the life cycle of the overall system being developed and maintained, and do not commit too early on necessary resources required to support the overall system.

HSI should guaranty that technology be adapted to people and not the opposite. Since HSI is strongly based on modeling and HITLS, using virtual prototypes, tangibility has become a key concept that requires deeper analysis. Mastering system's complexity, flexibility, stability, maturity and sustainability contributes to improving system's tangibility. Making tangible things requires both innovation and experience (two concepts, often considered as antagonist, that should be combined to make sense of what is unknown and known), and systematic testing. HSI is a goal in industry that requires more scientific attention.

This chapter proposes a system-of-systems structure-function approach combined with a cognitive-physical distinction, and more specifically on cognitive-physical function allocation among a society of agents.

References

- Ballerini, M., Cabibbo, N., Candelier, R., Cavagna, A., Cisbani, E., Giardina, I., Lecomte, V., Oriandi, A., Parisi, G., Procaccini, A., Viarle, M., Zdravkovic, V. (2007). Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study. *PNAS US National Academy of Sciences*. 105(4), 1232-1237.
- Boy, G.A. (book to appear). *FlexTech: Human-Centered Integration* of Increasingly-Autonomous Complex Systems.
- Boy, G.A. (2017). Human-centered design of complex systems: An experience-based approach. *Design Science Journal*. Cambridge University Press, U.K., Volume 3. https://www.cambridge.org/core/journals/design-science/article/humancentered-design-of-complex-systems-an-experiencebased-approach/F726100E2E7D0C825CB526FD7412C48E.
- Boy, G. A. (2016). *Tangible Interactive Systems: Grasping the Real World with Computers*. Springer, U.K. ISBN 978-3-319-30270-6.
- Boy, G.A. (2015). On the Complexity of Situation Awareness. *Proceedings 19th Triennial Congress of the IEA*, Melbourne, Australia, 9-14.
- Boy, G. A. (2013). *Orchestrating Human-Centered Design*. Springer, U.K. ISBN 978-1-4471-4338-3.
- Boy, G. A. (2011). (Ed.) *Handbook of Human-Machine Interaction: A Human-Centered Design Approach*. Ashgate, UK.
- Boy, G. A. (1998). *Cognitive Function Analysis*. Praeger/Ablex, USA; ISBN 9781567503777.
- DeLaurentis, D. (2005). Understanding Transportation as a System of Systems Design Problem. *43rd AIAA Aerospace Sciences Meeting Proceedings*. Reno, Nevada. AIAA-2005-0123.
- Endsley, M.R. (1995a). Measurement of situation awareness in dynamic systems. *Human Factors*, 37(1), pp. 65–84.
- Endsley, M. R. (1995b). Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *37*(1), pp. 32-64.
- Endsley, M.R. (1998). A comparative analysis of SAGAT and SART for evaluations of situation awareness. In *Proceedings of the Human Factors and Ergonomics Society 42nd Annual Meeting* (pp. 82–86). Santa Monica, CA: The Human Factors and Ergonomics Society.
- Endsley, M.R. & Garland, D.J. (Eds.) (2000). *Situation awareness analysis and measurement*. Mahwah, NJ: Lawrence Erlbaum Associates.
- Fitts, P.M. (Ed.) (1951). *Human engineering for an effective air navigation and traffic control system*. National Research Council, Washington, D.C.
- Investigation Report AX001-1-2/02 (2004). Bundesstelle für Flugun- falluntersuchung (BFU, German Federal Bureau of Aircraft Accident Investigation), Braunschweig, Germany, May.

- Kaptelinin, V. & Nardi, B. (2006). *Acting with Technology: Activity Theory and Interaction Design*. MIT Press, USA.
- Kuchar, J. K & Drumm, A. C. (2007). The Traffic Alert and Collision Avoidance System. *Lincoln Laboratory Journal*, 16(2), 277-296.
- Pemmaraju, V. (2013). 3 Simple Rules of Flocking Behaviors: Alignment, Cohesion, and Separation. Envatotuts game development tutorials (Retrieved on November 8, 2018: https://gamedevelopment.tutsplus.com/tutorials/3-simple-rules-of-flocking-behaviors-alignment-cohesion-and-separation--gamedev-3444).
- Potts, W. K. (1984). The chorus-line hypothesis of coordination in avian flocks. *Nature* 309, 24 May 1984, pp. 344-345.
- Reynold, C. W. (1987). Flocks, Herds, and Schools: A Distributed Behavioural Model. *Computer Graphics*, 21, July 1987, pp. 25-34.
- Schwaber, K. (1997). Scrum development process. In *OOPSLA Business Objects Design and Implementation Workshop Proceedings*, J. Sutherland, D. Patel, C. Casanave, J. Miller & G. Hollowell, Eds. Springer, London, U.K.
- Schwaber, K. (2004). *Agile Project Management with Scrum*. Microsoft Press. ISBN 978-0-7356-1993-7.
- Sutherland, J. (2014). *Scrum: The Art of Doing Twice the Work in Half the Time*. Crown Business. ISBN-13: 978-0385346450.