Human–Computer Interaction Series

Guy André Boy

Tangible Interactive Systems

Grasping the Real World with Computers

Springer

Human–Computer Interaction Series

HCI is a multidisciplinary field focused on human aspects of the development of computer technology. As computer-based technology becomes increasingly pervasive – not just in developed countries, but worldwide – the need to take a human-centered approach in the design and development of this technology becomes ever more important. For roughly 30 years now, researchers and practitioners in computational and behavioral sciences have worked to identify theory and practice that influences the direction of these technologies, and this diverse work makes up the field of human-computer interaction. Broadly speaking it includes the study of what technology might be able to do for people and how people might interact with the technology. The HCI series publishes books that advance the science and technology of developing systems which are both effective and satisfying for people in a wide variety of contexts. Titles focus on theoretical perspectives (such as formal approaches drawn from a variety of behavioral sciences), practical approaches (such as the techniques for effectively integrating user needs in system development), and social issues (such as the determinants of utility, usability and acceptability).

Titles published within the Human–Computer Interaction Series are included in Thomson Reuters' Book Citation Index, The DBLP Computer Science Bibliography and The HCI Bibliography.

More information about this series at http://www.springer.com/series/6033

Guy André Boy

Tangible Interactive Systems

Grasping the Real World with Computers

Springer

Guy André Boy
School of Human-Centered Design,
Innovation and Art
Florida Institute of Technology
Florida, USA

NASA Kennedy Space Center
Florida, USA

ISSN 1571-5035
Human–Computer Interaction Series
ISBN 978-3-319-30269-0 ISBN 978-3-319-30270-6 (eBook)
DOI 10.1007/978-3-319-30270-6

Library of Congress Control Number: 2016936565

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG Switzerland

Preface

"Are you a psychologist, a philosopher, or an engineer?" This is the question several people, who I met during the last three decades, repeatedly asked me. For sure, I have an engineering background, mostly automatic control, computer science, and aerospace engineering. Therefore, I am supposed to do tangible engineering. Incidentally in the early 1980s, I started working on the two-crewman cockpits and the fly-by-wire technology. I was research scientist at ONERA (the French Aerospace Research Administration) working with flight test pilots and engineers at Airbus.[1] The problem was for us to certify commercial transport aircraft with two crewmen instead of three. The FAA rules required us to take into account human factors. Very few specifications were provided in the FAA Rules PART 25 Appendix D (a little bit more than half a page—see P25-D in the reference list). They were described in the form of criteria for determining minimum flight crew, and more specifically 10 workload factors such as accessibility to flight instruments and controls, number, urgency, and complexity of operating procedures, mental and physical efforts, systems monitoring, degree of automation, communication and navigation workload, emergency workload management, and crewmember incapacitation.

We then studied human factors and ergonomics. Our first attempt was centered on physiological measures, such as ECG[2] and EEG.[3] Indeed, engineers liked "objective" measures! We could measure electric physiological signals coming from people, in this case pilots. However, the main problem was the interpretation of these signals. Interpretation is always subjective because it requires opinions and judgments from experts. Physiologists were able to provide models, but these models were very context dependent and did not provide meaningful-enough explanations of what was really going on when pilots were flying and, at the same time, trying to manage a failure recovery process (e.g., engine failure). We then decided to use methods that were based on subjective assessment from the start. A good example

---

[1] Called *Airbus Industrie* at that time.

[2] Electrocardiogram.

[3] Electroencephalogram.

is the Cooper-Harper workload assessment method that pilots had to learn in order to self-assess their own workload on a 1-to-10 subjective scale (Cooper and Harper 1969). Pilots' subjective assessments were correlated to workload subjective assessments of an observer located in the cockpit on the jump seat behind them. This approach worked perfectly. This was when I decided to formally attend psychology classes at the University of Toulouse. I learned cognitive psychology, got deeply interested in cognitive science and artificial intelligence, and actively participated in the making of cognitive engineering and human-centered design (HCD) since then (Boy 2003, 2011).

Tangibility: From the User Interface View to the Systems View

Thus, for the last four decades, I learned how and modestly contributed to articulate engineering with human and social sciences when designing life-critical systems,[4] more specifically aerospace systems. After a few years of practice, I understood that human factors were discovered too late during the life cycle of a product to be taken into account seriously. What human factors specialists produced for a long time was mostly informative for training, but cosmetic and not really effective for design. They contributed to the design of user interfaces and operational procedures to improve human adaptation to engineered systems. Since the 1980s, user interfaces became almost exclusively supported by computers and software. Consequently, Human-Computer Interaction (HCI) progressively became the inevitable solution to increasingly computerized Human-Machine Systems (HMS).

The **HCI-HMS distinction** is major. The early conference series on Human-Machine Interaction and Artificial Intelligence in Aerospace was renamed Human-Computer Interaction in Aeronautics (HCI-Aero) in 1998. The term "machine" was used in the HMS community to denote a mechanical system. Pilots were interacting with mechanical devices (e.g., maneuvering the yoke that was mechanically linked to the flight control surfaces of the aircraft). Today, when we talk about an **interactive cockpit**, we mean interacting with computers using a graphical "point-and-click" display system (i.e., pilots interact with a software that itself interacts with the flight control mechanisms). When I first heard about the term "interactive cockpits," I asked: "weren't aircraft cockpits interactive before?" Indeed, pilots were used to interact with mechanical surfaces of the aircraft directly and physically through mechanical control devices. Today with the fly-by-wire, pilots interact with computers that themselves interact with the mechanical systems of the aircraft.

---

[4] Life-critical systems (LCSs) are defined as an integrated set of people and interactive systems that have three main emerging properties, i.e., safety, efficiency and comfort. For example, aircraft, power plants, cars, hospitals, houses and cities are LCSs. New types of LCSs involve security because they are software-based networked and open to the world with little protection. For example, when you loose your smartphone, you realize how life-critical it is. We created this kind of LCS. Consequently, we also created the need to deal with their specific life-criticality. This is a complexity issue because emerging LCS issues are very hard to discovered before use time.

Therefore, the concept of interaction shifted from human-mechanical systems interaction to human-computer interaction.

Consequently, we need to be careful when we talk about interactivity. Human-computer interaction is not only a matter of cognition, but also a matter of tangibility. We are facing the paradox of information overload and information access as well as loosing the sense of physical things. This is why tangibility has become a central focus of current socio-technical evolution.

HCI was born during the early 1980s as a specific branch of computer science. The Special Interest Group on Computer-Human Interaction (SIGCHI) of the Association for Computing Machinery (ACM) created a famous conference series called CHI (Conference on Human Factors in Computing Systems). The first CHI conference was held in Gaithersburg, Maryland, USA in 1982. It was, and still is, a **design** conference more than anything else. CHI focuses on the design of human-centered computing systems (i.e., the whole thing is a computer, even cell phones that populated CHI conferences since the late 1990s). However, **domains**, such as nuclear power plants, aircraft, spacecraft, and other mechanical systems where software was progressively embedded, were never well integrated in the CHI community. For a long time, most HCI solutions and methods were developed with office automation in mind. Today, even if HCI diversified in various application domains, tangibility is still only focused on user interfaces, and not on large complex socio-technical systems.

The concept of user interface is already a concept of the past when it is considered as an add-on. I explained in my previous book (Boy 2013) that systems were designed and developed from **inside-out** (i.e., technological means are engineered without taking into account people who will use them; consequently, when they are fully developed, artifacts such as user interfaces and operational procedures need to be designed and developed). This was the necessary approach of the twentieth century, where engineered systems required human factors and ergonomics specialists to be usable. Since the beginning of the twentieth century, modeling and simulation capabilities enable the development of virtual prototypes that can be tested by appropriate end users. Consequently, systems can be designed from **outside-in** (i.e., usage purposes can be designed taking into account people who will use them from the beginning; people's activity can be tested; emerging behaviors can be discovered and seriously considered in the design process; user interfaces and operational procedures are integrating components of the system from the beginning). Inside-out engineering was about developing technological means requiring user interface development in the end to finally find out technology capabilities and usefulness purposes. Outside-in design is about purposes from the beginning and integration of appropriate technology to fulfill them, involving the participation of potential end users. This is what HCD is about.

User interface and automation are concepts of the twentieth-century bridging the gap between technology-centered engineering and users.

Virtual engineering and tangibility are concepts of twenty-first century bridging the gap between human-centered design, systems engineering, and people.

When I wrote this book, I also started to be in charge of the Human-Systems Integration Working Group of INCOSE.[5] Indeed, systems engineering and human-systems integration, coming from two different approaches (i.e., technology-centered engineering and HCD, respectively), developed independently. I was looking for an integration of these two approaches.

Many colleagues have encouraged and collaborated in my efforts to develop and demonstrate the value of tangibility in virtual engineering and human-centered design of current interactive systems. I am indebted to Mike Conroy, Ondrej Doule, Nikki Hoier, Christophe Kolski, Jason Miller, Jen Narkevicius, and Lucas Stephane for active discussions and provision of helpful comments on this integration. I also want to thank all my students and research scientists at the School of Human-Centered Design, Innovation and Art of Florida Institute of Technology, and NASA Kennedy Space Center who directly or indirectly helped me in shaping the concept of tangible interactive systems developed in this book.

Melbourne, FL, USA Guy André Boy
December 6, 2015

References

Boy GA (2003) (ed) L'Ingénierie Cognitive: interaction Homme-Machine et Cognition (The French handbook of cognitive engineering). Hermes Sciences, Lavoisier

Boy GA (2011) (ed) Handbook of human-machine interaction: a human-centered design approach. Ashgate, UK. ISBN 978-0-7546-7580-8

Boy GA (2013) Orchestrating human-centered design. Springer, London

Cooper G, Harper R (1969) The use of pilot rating in the evaluation of aircraft handling qualities. Technical report TN D-5153, NASA

P25-D. Part 25: airworthiness standards: transport category airplanes, Appendix D to Part 25, page 163. Docket No. 5066, 29 FR 18291, 24 Dec 1964; as amended by Amdt. 25–3, 30 FR 6067, 29 Apr 1965

---

[5] International Council on Systems Engineering.

Contents

Chapter 1
Introduction

> *The real voyage of discovery consists not in seeking new landscapes, but in seeing with new eyes. – Marcel Proust*[1]

Hiroshi Ishii was the first human-computer interaction (HCI) scientist to design seamless interfaces between humans, digital information, and the physical environment. He and his team were seeking to change the "painted bits" of graphical user interfaces to "tangible bits" by giving physical form to digital information (Ishii and Ullmer 1997). We need to **distinguish** between this purely HCI concept of **tangible user interfaces** (TUIs)[2] (Ullmer and Ishii 2001) and **tangible interactive systems** (TISs) taking into account the tangibility of systems and not only the user interface enabling interaction with digital information through the physical environment. In addition, TISs are presented as a grounding concept for **human-centered design (HCD) and systems engineering**. The concept of TIS goes far beyond the concept of TUI and addresses large complex systems. Within the context of aerospace complex systems design and management, I recently proposed the shift from automation to tangible interactive objects (Boy 2014). In fact, the concept of "system" is more appropriate than the concept of "object" because it encapsulates objects, processes, and people. Systems can be abstract or concrete and have functions and structures; they include software and hardware. In addition, HCD that puts humans at the center of the design process differs from traditional human factors and ergonomics (HFE) that are commonly taken into account after the engineering process.

---

[1] This quote comes from "A la recherche du temps perdu" (In search for lost time), a novel in seven volumes by Marcel Proust (1871–1922). This work was published in France between 1913 and 1927 (first volume by the Grasset Publishers and then by many others).

[2] https://en.wikipedia.org/wiki/Tangible_user_interface

G.A. Boy, *Tangible Interactive Systems*, Human–Computer Interaction Series,
DOI 10.1007/978-3-319-30270-6_1

Automation, Activity, and Cognitive Functions

We created and are still creating systems that have very sophisticated functions, which were previously handled by people. In Aeronautics, **automation** was, and still is, typically developed incrementally by accumulating layers upon layers of software—progressively isolating pilots from actual mechanical systems. This approach requires constant revisions and repairs to most often adapt people to systems and rarely systems to people. Instead of repairing when it is too late, it is always better to incrementally improve solutions at design time. For this reason, an HCD environment should be available where **both end users and designers learn from each other**. On one side, designers should learn what end users (e.g., pilots) need, can and cannot do in order to design and develop appropriate technology (e.g., aircraft). On the other side, end users should learn how to use new technology. Today, end users need to understand and practice software-intensive systems management (i.e., a very new endeavor in the history of humanity). Pathways for feedback should be provided to end users for continuous improvement to take place.

During the mid-1990s, when I was studying aircraft automation, I realized that it was important to develop the **cognitive function** representation to analyze function allocation between humans and life-critical systems. From this perspective, this book is a follow-up of my 1998 book on Cognitive Function Analysis (CFA), when I tried to understand and rationalize the relationship between **task** (i.e., what is prescribed to the human operator) and **activity**[3] (i.e., what this human operator can effectively do when he or she executes the task). I decided then to call this relationship between task (i.e., an input) and activity (i.e., an output), a "cognitive function." CFA was developed within the context of aircraft cockpit automation (Boy 1998a, b) and based on the concept of agent. We will consider that an agent is a system that is capable of acting on its environment. An agent has at least one cognitive function.

Tasks are prescriptions that people use to do things.

Activities are what people really do when they execute tasks.

Cognitive functions are processes that enable people to transform tasks into activities.

---

[3] People's activity is taken in the ethnomethodology sense (i.e., what people actually do) and not on tasks (i.e., what people are prescribed to do). This distinction between task and activity was already described to define the cognitive function representation for the implementation of cognitive function analysis (Boy 1998b, 2011, 2013), as well as activity theory (Leont'ev 1981; Kaptelinin 1995). The concept of activity is related to Ochanine's concept of operative image (Paris I Seminar on D. Ochanine's Operative Image 1981). More recently, exploring the social aspects of interactive systems, Paul Dourish proposed the foundations of a phenomenological approach to human-computer interaction through embodied interaction (Dourish 2004). The concept of activity has then to be understood as both cognitive and embodied.

In the past, activity analysis was only carried out on existing, but not yet formalized, **practice**.[4] Applied to actual design and engineering work, this kind of approach reinforced what people were doing with old technology, and inference of possible practice using new technology was somehow deemed hazardous. Of course, the value of analyzing current activity led to considering **continuity** between current and new practices. It also created some kind of evolution viscosity and in some cases prevented radical changes, precisely for continuity purposes.

Indeed, the difficulty in the use of CFA is in the observation of activity. You must have a fully developed system to observe what people can do using it. Until recently, this was only possible at the end of the manufacturing process (e.g., during flight tests in aeronautics). Today, things have changed. We have simulation capabilities that enable us to test the system being designed very early during the design process. Consequently, activity can be observed and further analyzed. Therefore, CFA has become possible at design time. This is great progress because we can test both deliberatively designed and emerging cognitive functions.

Cognitive functions are cognitive in the sense that they are capable of information processing. Consequently, **automation** can be viewed as cognitive function transfer from humans to systems. This kind of transfer has been incremental during the last decades of the twentieth century. Of course, a system cognitive function (SCF) is different from an equivalent human cognitive function (HCF). They are not generated in the same way (i.e., the software engineering process of making SCFs is certainly very different from the human cognitive process of learning HCFs). However, they both should have (or must have) a similar **role**, a similar **context** of validity, and a similar set of allocated **resources** that enable their execution (Boy 1998b). Resources can be physical or cognitive. At this point, we can see that a resource can be a cognitive function. Therefore, the concept of cognitive function is recursive. This is also true for a physical function. The irony is that when they are well designed and manufactured, SCFs can be more effective and reliable than their equivalent HCFs, when they are of course used within their context of validity. In this book, we will see how cognition plays an important role in the interaction between humans and software-based systems and how we need to better understand the orchestration of human and system cognitive functions. Cognition however is not the only factor that we need to consider; emotions and social factors are also crucial in this framework.

Human and system cognitive functions are both represented by a role, a context of validity, and resources that enable their execution.

---

[4] The term "practice" is used in this book to talk about activity. This interpretation is based on practice theory. Practice theory focuses on how people's purposes and intentions contribute to shape and change their environment. It is strongly based on Pierre Bourdieu's sociological work (Bourdieu 1980), and more specifically the notion of "habitus" (i.e., permanent embodiment of social order). Practice theory is intimately related to ethnography. I will use the same meaning when I will introduce the "maturity of practice" concept.

From Automation to Tangible Interactive Systems

Back in the 1980s, when conventional fly-by-the-seat-of-your-pants technology was replaced by fly-by-wire technology in commercial aircraft, pilots had difficulties using innovative onboard computing systems, such as the Flight Management System (FMS), doing several flying tasks for them. They had to learn how to delegate—not an easy task! When you delegate, you do not control all tasks at the finest grain level. Pilots incrementally adapted to this new way of thinking and doing things. This computerization evolution never stopped and is still not ready to stop. Today, there are two main issues. The first issue is related to **complacency** due to good performance and reliability of these systems, hence new habits of pilots to trust them (often too much trust nowadays). When an unexpected situation occurs, pilots may not be in the control loop any longer—we talk about lack of situation awareness and control awareness. The second issue is related to the **virtual** life induced by the use of these systems—pilots need to learn how to deal with the real world through computers. Unlike pilots of the 1980s who were used to dealing with physics and mechanics, and (some of them) hated computers, young pilots of today are more familiar with virtual worlds (i.e., they are coming from the computer-game generation) and may have some problems dealing with real-world events that include physics. **Tangibility** becomes a major issue. Something tangible is, by definition, capable to be physically graspable, touched, or discernible by the touch. It is also understood as real or actual, rather than imaginary or visionary.[5] Tangible can be thought as the opposite of vague. In other words, flying highly automated aircraft requires understanding of classical flight handling qualities and flying tasks handled by onboard systems. Today, people need to be able to **grasp the real world using computers**.

Let's take the example of *TangiSense* (Kubicki et al. 2013; Lebrun et al. 2015), which consists in a set of tangible objects, considered as agents, moving or moveable on an interactive tabletop. Each tangible object has a behavior and a role. TangiSense was used to support the simulation of road traffic management. Authors, located around TangiSense interactive tabletop, were able to interact with other people (decision-makers) in a simultaneous and collaborative way during a simulation session.

During the twentieth century, we automated mechanical machines, i.e., we incorporated software into hardware (Fig. 1.1). This automation was done because we could do it (i.e., human-centered purposes were too often investigated once technological means were already developed).

Many research efforts were developed to investigate benefits and drawbacks of automation (Bainbridge 1983; Billings 1991; Sarter et al. 1997). Since the beginning of the twenty-first century, we have been designing systems by developing software programs that lead to human-in-the-loop simulations, which enable us to test various kinds of purposes. Today, the question is less investigating drawbacks of automation as we did during the last decades of the twentieth century, than discovering emerging properties of **Tangible Interactive Systems,** or **TISs** (Fig. 1.2).

---

[5] http://dictionary.reference.com/browse/tangible

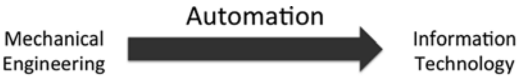


Fig. 1.1 Automation as the "old" shift from mechanical engineering to information technology: the inside-out approach (Boy 2014)

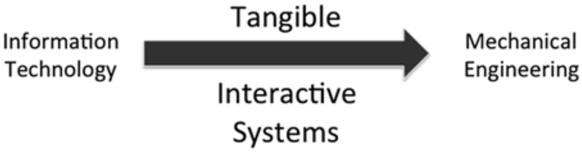


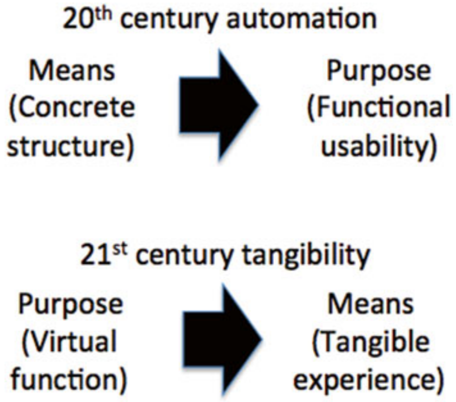
Fig. 1.2 Tangible interactive systems as the "new" shift from information technology to mechanical engineering: the outside-in approach (Boy 2014)

The concept of tangible interactive system extends the concept of tangible user interface already developed in human-computer interaction. TISs are useful bricks for **virtual systems engineering** (VSE). VSE is based on modeling and simulation. It enables human-in-the-loop simulations (i.e., VSE support human-centered design). Let's take the office automation revolution example, which drastically changed cognitive functions dealing with office tasks. Until the end of the 1970s, computers were big machines used by a few specialists in very constrained environments. The development of **microcomputers** started to democratize information technology at work and at home. The first leading type of application became word (text) processing. When I prepared a PhD in the end of the 1970s, I wrote my dissertation by hand and, as everybody else in this situation did, I had the secretary of my department type it. Nobody at that time thought that we would type our research reports or dissertations by ourselves. Today, we all use our favorite text processing applications and hardly think that someone else would type it for us. Computer-supported writing cognitive functions have become sustainable, and the role of dactylograph secretaries obsolete. Text processing became tangible because it enabled us to easily produce information content, share it, and use it as a collaborative support system thanks to email and Internet.

A major distinction: tangible interactive systems are in virtual systems engineering what tangible user interfaces are in human-computer interaction.

Virtual systems engineering is model based and supports human-in-the-loop simulations.

Fig. 1.3 Paradigm shift: from “means-to-purpose” to “purpose-to-means”



More recently, mobile devices were developed. New cognitive functions emerged to satisfy the need of always being connected. Several categories of information transfer were developed such as texting, emailing, and speaking directly. Smartphones are excellent TIS examples, which are integrated systems providing such capabilities.

One of my graduate students proposed that HCD should be called Human-Centered Engineering. She said: “When I tell people my field of study, I get very similar reactions – what they hear is ‘design’ and assume it is like ‘Interior Design’ for example. This becomes most evident especially when a person is pseudo-listening, which many people do – you pick out the words you know or remember the last word spoken… It seems I have to explain *what* my field is before I can even explain what I am even doing in the field itself.” My reaction was to say that this debate is indeed purposeful and, yes, we need to explain what HCD is about.

The distinction between HCD and Technology-Centered Engineering (TCE) should be put forward. HCD goes **from purpose to means** (i.e., the architect view), instead of traditional engineering that mostly goes from means to purpose (i.e., the builder/mason view). This shift from twentieth-century automation issues that attempted to provide functional usability to concrete structures to twenty-first-century tangibility that requires looking for tangible experience of human-centered designed virtual functions, as illustrated in Fig. 1.3.

The Authority Issue: Being in Charge and Accountable

TISs have cognitive functions that now can be identified early on during the design process. These cognitive functions can be either deliberately chosen to satisfy the purpose of the system being designed or discovered as emergent properties of the system being used. All these cognitive functions can also be characterized in terms of **control, responsibility, and accountability**. TISs can provide tremendous power to their users. Spreadsheets, for example, are TISs that provide an enormous amount of power to finance specialists. When spreadsheets and connectivity became available and stable, top managers suddenly were able to financially control their various enterprises safely, efficiently, and comfortably, like pilots control their airplanes in their cockpits. Consequently, top managers and stakeholders learned how to master these technologies to control companies and institutions by using a single independent variable, which is money! They enabled reinforcement and stabilization of finance-driven management. However, there are other independent variables that have to be considered if we want to develop and maintain a human-centered philosophy in socio-technical systems. For example, these independent variables can be well-being, health, liberty, and pursuit of happiness.

Designing and using tangible interactive systems involves control, responsibility, and accountability.

For example, Volkswagen is the most profitable automobile manufacturer in the world, which maintained a solid reputation of reliability until the International Council for Clean Transportation (ICCT), a non-governmental organization specialized in clean transportation, decided to test gas emissions of some diesel cars in the United States. ICCT requested the support of the Center for Alternative Fuels Engines and Emissions of West Virginia University to perform the tests. *The Economist* (September 26, 2015) reports that "emissions of nitrogen oxides (NOx) and other nasties from cars' and lorries' exhausts cause large numbers of early deaths—perhaps 58,000 a year in America alone, one study suggests. So the scandal that has engulfed Volkswagen (VW) this week is no minor misdemeanor or victimless crime… The German carmaker has admitted that it installed software on 11 million of its diesel cars worldwide, which allowed them to pass America's stringent NOx-emissions tests. But once the cars were out of the laboratory the software deactivated their emission controls, and they began to spew out fumes at up to 40 times the permitted level." This kind of practice is of course illegal, but more fundamentally VW main motivation, purpose, and goal were financial and not human centered.

In the Orchestra framework sense (Boy 2013), current socio-technical world's music theory is centered on the single money dimension. This is why we see many catastrophes, in the **complexity** science sense (i.e., in René Thom's catastrophe theory sense—Thom 1976). Indeed, complexity science tells us that when we project a multidimensional space onto a space of lower dimensions, a fortiori one dimension, we should expect **catastrophes**. For example, we have projected our multidimensional world (e.g., that includes human rights variables such as life, liberty, and pursuit of happiness) onto a single dimension world of finance (i.e., money being the single variable). HCD, as a philosophy and a set of methods and tools trying to harmonize technology, organizations, and people, has the mission to bring back all these variables that make our **humanity**[6] more livable, sustainable, and richer in the human sense. Therefore, in addition to being visionary, human-centered designers need to be responsible. Indeed, HCD is about design and use. Therefore, human-centered designers should investigate the various kinds of use of, and possible futures induced by, systems that they design.

Creativity and Emergence Rationalization

HCD is about **creativity**, in the sense of **synthesis** and **integration**. HCD is prior to engineering and should lead engineering. The main reason HCD is possible today, and not before, is because we have modeling and simulation tools that enable it to be done. We can work with product's actors (e.g., designers, end users, maintainers, trainers, and so on) because we can work on very realistic models and simulations from the start and carry out effective participatory design. In other words, modern computing means are crucial.

Software and computer networks have created new types of **complexity** in our everyday life. The best example is the Internet. However, connectivity will quickly expand far beyond the Internet, as we know it today. For example, the rapid growth of aircraft density in some zones of airspace requires special attention (i.e., 4.5 % growth per year average for the last 30 years). Current investigations show that we should anticipate a regulated growth instead of a global growth as we typically thought during the last decade (Challenge of Growth, 2013[7]). This means there could be a decrease of growth rate, and EUROCONTROL Statistical Reference Area (ESRA) already calculated a 1.8 % average annual growth during the period 2012–2035. In any case, some big airports are already saturated and require an integrated approach to solving the problem of delays. For example, the concept of New York Integrated Control Complex (NYICC) was proposed for improving the efficiency of operations by "integrating terminal and en-route airspace to expand the use of 3-mile separation procedures and improve communication and coordination" (FAA 2007). This kind of process integration involves creativity and complexity analysis. The combined problem of density, capacity, and safety requires search for solutions on the ground and onboard airplanes. There are basically three solutions to this problem. First, we could build more airports. Economy and ecology currently tend to dictate not to do this. Second, we can build bigger commercial aircraft. Airbus took the lead and built the A380, capable of transporting up to 800 passengers. The small number of these aircraft is far from solving the problem yet. Third, we could automate the sky! This means that instead of keeping implicit connectivity among aircraft, we could make it explicit (i.e., making each aircraft aware of the

---

[6] Humanity should be understood in the sense of human condition, that is human existence in harmony with nature and our growing sociotechnical world.

[7] http://www.eurocontrol.int/sites/default/files/article/content/documents/official-documents/reports/201306-challenges-of-growth-2013-task-4.pdf

traffic around it and developing capabilities that keep the overall airspace safe). This third solution is currently studied within programs such as NextGen[8] in the United States and SESAR[9] in Europe. When we analyze this new problem, it turns out that the current centralized air traffic control system is very limited for the management of high-density traffic. Therefore, a decentralized solution should be found. This means that aircraft flows should be considered as flocks of birds where each bird is aware of the presence of the other birds around it and act accordingly. Consequently, each aircraft should be equipped with specific cognitive functions that automatically detect other aircraft around it, infer appropriate actions, and act appropriately. There are new types of **multi-agent** complex systems. Understanding of the complexity of **Human-Systems Integration** (HSI) in such highly dynamic multi-agent environment can be supported by use of the cognitive function representation that enables better understanding of **emergent behaviors and properties**. An example of cognitive function orchestration will be provided in this book when the association of HCD and systems engineering are described in Chap. 3.

Creativity is about synthesis and integration.

Designing a new TIS is about understanding the complexity of induced human-systems integration through the discovery of emergent behaviors and properties using HITLS.

Software prototypes can be developed very early during the design process. Consequently, usefulness and usability tests are possible during the design process also (as opposed to traditional human factor studies when products are fully developed). This is a radical shift. These prototypes enable us to co-adapt technology to people and organizations, at least functionally speaking. More specifically, **human-in-the-loop simulations** (HITLS) are now possible using very realistic simulated environments, which enable observation of people using systems being designed, and eventually discovery of emerging patterns and properties (Fig. 1.4).

More specifically, the use of CFA combined with HITLS enables us to discover emergent cognitive functions. During the early 1980s, the main issue was the difficult adaptation of people to information technology and its integration in various domains (e.g., pilots adaptation to new generation of fly-by-wire aircraft). I remember old pilots trying to adapt to the use of Cathode Ray Tubes (CRT) displays and later to the Flight Management System (FMS). Today, technology has evolved to present different challenges to the end user. People know more about information technology (IT) in their everyday life, and the main issue is adaption in our real physical world using IT. Young people do not generally seem to have difficulty living in the virtual world provided by Internet, social networks, smart phones, and

---

[8] Next Generation Air Transportation System.

[9] Single European Sky Air Traffic Management Research.