Chapter 2
Looking for Tangibility

Today, drones (i.e., autonomous or remotely piloted flying machines) are being built and start to be used. Military operations using drones are now becoming better established. However, drones are still under-regulated in the civilian world. Main questions are about safety and security. Therefore, drone operations have to be seriously considered by commercial and general aviation regulatory institutions, such as ICAO[1] at the international level, FAA[2] in the USA, and DGAC[3] in France. More generally, this kind of issue always comes into play when new life-critical technology is integrated in our everyday life. In other words, this new technology has to be tangible for use (i.e., we need to demonstrate that use of technology makes sense from various relevant points of view).

Something is tangible when it is **graspable** in the **physical** sense, but also **believable** in the **figurative** sense (e.g., an idea or a concept that cannot be grasped by the mind). In the case of drones, they will become operationally tangible when they will be safely, efficiently, and comfortably usable and useful for tasks, such as package delivery, image capture for news purposes (e.g., television), and disaster management support. In addition, new technology brings new properties that need to be explored. For example, drones become birds that can see what people cannot see. They create new supports to situation awareness and other functions that were impossible before. In addition to regulatory framework (the safety issue), important concerns that need to be taken into account are privacy (the ethical issue), unfair competition (the economic development penalization issue), and the toy effect (the use-for-fun issue). Technology design has shifted from mechanical engineering to information technology. For example, mechanical and aerospace engineers created and developed aircraft the way they are today; information technology specialists, who do not have professional aviation training, already started to be the designers and developers of a new generation of drones (e.g., for 3D movie making). Some of

[1] International Civil Aviation Organization.

[2] Federal Aviation Administration.

[3] Direction Générale de l'Aviation Civile.

G.A. Boy, *Tangible Interactive Systems*, Human–Computer Interaction Series,
DOI 10.1007/978-3-319-30270-6_2

the drones are 3D printed! As already stated before, making tangible software-based things becomes the prominent issue over the previous automation issue brought by the twentieth century engineering practice. Therefore, new certification rules need to be developed.

Tangibility can be physical and/or figurative.

Systems become tangible when they can be used safely, efficiently, and comfortably.

(HITLSs) and, ultimately, human-centered design (HCD). On a broader perspective, combined with systems engineering (SE), HCD is currently maturing as useful and effective support for human-systems integration (HSI) (Boy and Narkevicius 2013). More specifically, TIS design and development highly contribute to HSI and conversely!

A Short History of Automation

This section addresses the mutual influence of engineering sciences and life sciences, more specifically regarding the difficult question of understanding automation processes, as well as control, regulation, and autonomy concepts.

In the end of the eighteenth century, James Watt, a Scottish mechanical engineer, invented the steam engine and, most importantly for the sake of the content of this book, a centrifugal flyball governor, which proportionally controls engine speed by regulating the amount of fuel admitted into an engine. Watt offered the world one of the first automatons based on mechanical engineering theory and practice. Mechanical engineering started to really develop during the nineteenth century, with the development of steam machines and trains in particular. It was essentially technology centered until the end of the twentieth century. However, machines that were produced needed to be controlled. Consequently, control mechanisms and theories were developed.

In the end of the nineteenth century, Claude Bernard, a French physiologist,

The automation concept constantly swapped from hard sciences to life and social sciences back and forth.

coined the term *milieu intérieur* (i.e., internal environment), which Walter Cannon[4] named “homeostasis” later on. The regulating process insures consistency of the internal environment of the human body and preserves it from external aggressions.

[4] The term “homeostasis” was coined by Walter Cannon in 1930, referring to any process that living organisms use to actively maintain stable conditions necessary for survival (Cannon 1932).

It is considered as the main continuous compensating process, which maintains life of the whole body.

It is interesting to observe that, working totally independently, Watt and Bernard produced scientific results that led to a common theoretical and practical field currently called automatic control or automation. These two tracks (i.e., engineering sciences and life sciences) required more than one and a half century to reconcile and give a consistent theory of control.

In the first part of the twentieth century, Arturo Rosenblueth Stearns, a Mexican physician and physiologist and Cannon's student at Harvard University, continued to develop homeostasis and wrote papers together with Norbert Wiener, the father of cybernetics (Rosenblueth and Wiener 1945). They greatly contributed to explain the feedback control loop of regulated systems. They showed that combining life sciences and mathematics was a powerful mix. In this case, their joint enterprise led to the formalization of mathematical control theories and more specifically automatic control. This kind of models enabled later development in industry and research. Aircraft autopilots were rationalized using these theories.

More generally, human sciences and STEM[5] never stopped developing together. Until the 1980s, autopilots were developed using electric and electronics components. Then software started to become prominent, and a new kind of automation emerged. We were able to design and develop more complex algorithms. During the mid-1980s, the flight management system (FMS) was developed on top of autopilots to handle aircraft navigation automatically. The pilot's job shifted from control of flying quality parameters to management of avionics systems: a big step!

The era of mechanical/physiological control, even assisted by an autopilot, shifted to a new era of **cognitive management** of artificial agents.[6] Designing and developing these new software-intensive systems, we needed to rethink the classical validation and certification rules and invent new ones more appropriate for operational tests for these new information-based technologies. Cognitive psychology and cognitive anthropology became important reference disciplines for studying human-computer interaction, which led to cognitive engineering.

In the beginning of the 1980s, office automation started to penetrate our everyday lives. Text processors were developed and massively used to the point that jobs drastically changed. Until the end of the 1970s, we were using paper and pencil to write letters, documents, and reports. Some used typewriters. Most people did not

---

[5] Science, Technology, Engineering, and Mathematics.

[6] During the 1980s, computer science strongly developed and led to the extension of two new disciplines that are artificial intelligence (AI) and human-computer interaction (HCI). Creating the term "artificial intelligence" in 1955, John McCarthy and Marvin Minsky wanted to denote the science and engineering of making intelligent machines. An intelligent system, sometimes called an intelligent agent, should be able to perceive external signals, process them, and act on its environment with respect to three main principles that are safety, efficiency, and comfort. AI has the long-term vision of designing and developing intelligent robots capable of some degrees of autonomy. Human-computer interaction (HCI) was more short term, focusing on user interfaces and usability of systems. HCI led to deeper scientific developments of the concept of interaction design (we will see this later in this chapter).

think about typing their own text. There were people assigned for professional typing. A few years after, almost all of us typed, copied, and pasted our own texts and were reluctant to delegate this task to someone else. In addition to text processing, office automation led to the development of spreadsheets, computer-supported cooperative work, the Internet, and more recently sociomedia. People adopted these technologies because they became tangible (i.e., they are purposeful, usable, and useful in the society where we live).

From Control to Management

Basic automata exist for a long time. The clock is probably the oldest one. Air conditioning was used in ancient Egypt. Egyptians understood that the process of evaporation was producing negative enthalpy, and they exploited it by hanging moistened rugs in a location where there was enough air circulation to generate this evaporation process. More recently, mechanical and electromechanical refrigeration was invented. The thermostat[7] was developed as an **automatic control** system, which has a set point and an output. The output (e.g., the temperature of the room to be cooled) is sensed and compared to the set point (i.e., the temperature that you assigned on the thermostat), and the difference between the output and the set point minimized. This regulation principle is generic for all electrical regulators. Of course, there are various types of homeostatic processes that enable to minimize the difference between the output and the set point.

During the twentieth century, many automated machines were developed, starting by the washing machine, refrigerator, and automated transmission on cars. People have adapted to these kinds of automation now. In the beginning of the 1980s, a new type of automation started to be developed, supervisory control. Thomas Sheridan, Professor at MIT in the Department of Mechanical Engineering and Department of Aeronautics and Astronautics, coined the term of **supervisory control** to denote a process that involves several basic controllers whether they are humans or machines (Sheridan 1984). Supervisory control contributed to shift human work from manual control and basic automatic control to (automated) systems management. Human operators had to move from doing to thinking (i.e., **from control to management**).

[7] The term "thermostat" includes two concepts: the concept of "thermos" (i.e., θερμός, in Greek) and the concept of "statos" or stationary (i.e., στατός).

From Analog Signal Processing to Digital Computing

Up to the 1980s, electrical engineering and electronics governed automation. We were in the analog era. Signal processing strongly developed and was extensively used, leading to modern mechatronics today. A major revolution happened during the 1980s; digital computers, and more specifically microcomputers, became operational and massively used. Automation became digital. We entered into the software era. Programming languages evolved toward object-oriented programming, leading to both declarative and procedural programming. Design and development of systems progressively involved more cognitive skills. This is another reason of the emergence of cognitive engineering.

It became very easy to program software supporting the development of new automated systems, sometimes too easy! Software engineering had to keep up with the mandatory certification issues required for safety, efficiency, and comfort in life-critical systems (LCSs). Robust, resilient, and reliable software engineering methods are needed to handle the development of large software-based LCSs. Systems engineering developed such methods and keeps doing so. SCRUM, for example, is a very useful method that enables several design teams to reach a common goal by incorporating “the concepts of continuous improvement and minimum viable products to get immediate feedback from customers, rather than waiting until a project is finished” (Sutherland 2014). It is based on a holistic approach that belongs to the **agile** software development philosophy,[8] which is symbiotic to the HCD approach. It is based on the fact that requirements may change during the development process, and flexibility is required to incrementally modify both high-level and low-level developments. One of the basic assumptions is that the final product is defined incrementally. Agile design and development will be further presented in Chap. 6.

The evolution from signal processing to digital computing induced the emergence of the shift from the control of a machine to the management of systems.

From User Interface Ergonomics to Interaction Design

For a long time, machines were developed **from inside out** (e.g., engine, chassis, and car body were developed before drivers could start using the integrated product). Users were taken into account after technology was developed. This

[8] The Manifesto for Agile Software Development (http://www.agilemanifesto.org) has been written to improve the development of software. It values more individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change (flexibility) over following a plan (rigidity).

philosophy led to the concept of user interface. Indeed, once a machine was developed, a user interface was necessary for the users to operate this machine. The more complex the machine was, the more complex the user interface had to be.

Consequently, human factors and ergonomics (HFE) specialists took the job of designing and developing user interfaces. However, user interfaces cannot compensate all design flaws regarding operational issues. More specifically, when a machine is not fully autonomous, some kinds of (more or less) specialized people have to control or manage it. This engineering/HFE approach is still active today. It requires adaptation: adaptation of the machine to people (i.e., developing a good user interface) and adaptation of people to the machine (i.e., training people). This philosophy is technology centered and necessarily requires iterative adaptation of both humans and user interfaces because machine core technology has already been developed and cannot be, or can be slightly, modified for heavy-financial investment reasons. This approach typically leads to conflicts between engineers and HFE specialists, mostly because there is no real mutual understanding and constructive discussion between them.

The user interface issue emerged from the fact that up to the end of the twentieth century, technology had to be developed from scratch almost all the time. Therefore, engineers had to develop their ideas, try solutions, and could not provide a product to potential test users sooner than after its full development. There was not much choice!

Technology-centered engineering is based on an inside-out approach that starts by developing technology and discover human factors and ergonomics issues when technology is developed.

Human-centered design is based on an outside-in approach that starts by modeling and simulating technology in its environment together with involved stakeholders (formative evaluations) and incrementally continues by developing TISs.

Today, the situation is very different. Information technology provides tools and methods, which enable design and development of software-based prototypes that can be tested in HITLS environment very early on during the design process. Current M&S capabilities change everything. In particular, design can be done **from outside in** (e.g., virtual engine, chassis, and car body can be integrated at design time, and the overall virtual integrated product prototype can be tested by real users). End users can be taken into account before technology is developed.

Digital prototypes enable "interaction design" (Bolter and Gromala 2008). Interaction design is often defined as a process that consists in shaping digital things for people's use.[9] Interaction design is deeply rooted in the human-computer interaction (HCI) community. However, it takes insights and techniques from architecture, industrial design, cognitive science, social sciences, and, of course, computer science.

[9] https://www.interaction-design.org/

From Interaction Design to Human-Systems Integration

The HCI community started to develop in the early 1980s with the emergence of personal computers and more specifically text processing and office automation (Card et al. 1983; Norman and Draper 1986; Winograd and Flores 1987). User interfaces became digital. HCI started during a period when artificial intelligence (AI) was at its apogee (i.e., AI was very strong, even if it was much too ambitious at that time). HCI was more short term than AI and became a discipline on its own right; HCI and AI were always distinct disciplines (Grudin 2006). This is unfortunate because one can bring to the other and conversely. HCI is centered on interaction design, and AI is centered on content automation, reasoning, and machine learning.

In addition, the HCI community developed itself around user-centered design of friendly computers, which took various tangible forms (e.g., laptops, tablets, and smartphones). The question of tangibility was reduced to the use of a computer targeted toward very well-formatted tasks (e.g., text processing, spreadsheet-based calculation, drawing, telephoning, texting, and so on). Ishii and his team at MIT coined the term "tangible bits" to denote graspable and manipulable everyday physical objects. "The goal of Tangible Bits is to bridge the gaps between both cyberspace and the physical environment, as well as the foreground and background of human activities" (Ishii and Ullmer 1997).

Using computers as user interfaces (i.e., using HCI) and internal control mechanisms (i.e., using AI and control theories) for larger industrial systems, such as aircraft, spacecraft, hospital operating rooms, and nuclear power plants, leads to different kinds of problems. Consequently, specialized communities were created and developed, such as the HCI-Aero community (human-computer interaction in aerospace). In the beginning, HCI-Aero conferences emphasized user interfaces, computer graphics for aircraft cockpits and air traffic control workstations, and other HCI issues and techniques. The shift from inside-out engineering to outside-in design induced a new shift from HCI to **human-systems integration** (HSI), which is the HCI-Aero 2016 conference topic.

The concept of "computer," in terms of software and hardware, is embedded in the concept of "system." System has to be thought within the framework of the TOP (technology, organizations, and people) model (Boy 2013). The concept of "integration" has become fundamental in design and encapsulates the concept of "interaction." The HCI community studied the latter and came up with the already mentioned crucial concept and process of "interaction design" (Rogers et al. 2011). The systems engineering community, the International Council on Systems Engineering (INCOSE) in particular, is currently developing the concept and process of HSI (Boy and Narkevicius 2013). This book emphasizes the shift from HCI to HSI and associated design, integration, and management of tangible interactive systems.

The challenge today is the integration of cyberspace and physical systems into socio-technical environments (Rajkumar et al. 2010). Several attempts are currently developed. They deserve to be compared and further homogenized. The

cyber-physical system (CPS) approach that focuses on embedded systems requires extending the emphasis on human-systems integration. Human-centered design of CPSs has become a contemporary problem that needs to be properly addressed and solved. Again, the main issue is **integration**. We have technology, lots of technology! For example, cost and size of sensors are going down every day as their capabilities are going up. Computing technology is cheaper, more powerful, and more effective in terms of capabilities and dimensions. We now live immersed in the cloud!

Progress is made on alternative energy production, even if this sector needs to be boosted. We now need to work on large-scale problems that our planet Earth poses to us. For example, since most people are still massively migrating to cities, what will be the city of the future? How can we integrate transportation systems? What will be the lower-energy house of the future? Again, this is a problem of integration of existing technology and integration of systems to be created and developed in the sense of the TOP model. Several concepts emerge and are required to be properly addressed such as innovation, complexity, flexibility, maturity, stability, and sustainability (Chaps. 4, 5, 6, 7, 8, and 9 of this book). CPSs attack control and management of life-critical systems (LCSs). Examples of LCS are integrated systems in cities, transportation, health care, crisis management, and alternative energy management and education. This is the reason why HCDi, and now the School of Human-Centered Design, Innovation and Art, were created and developed at Florida Institute of Technology.

The Philosophical Shift from Mechanical Engineering to Information Technology

During the whole twentieth century, mechanical engineering was the top engineering discipline until information technology took progressively the lead. Software incrementally invaded mechanical things at the end of the last century. For example, when I was studying at the university during the 1970s, I was able to repair the engine of my car without any major issues. Today, it is impossible without going to the garage where a garage operator will test the car engine with a diagnostic system, which will display how much the repair will cost! He or she will read the technical reason on his/her computer screen to tell you what it means, just in case you ask! Mechanical parts have been categorized and standardized. This achievement considerably increased structural integration.

However, human-systems integration (HSI) remains a problem. Why? We need to acknowledge the philosophical shift from mechanical engineering to information technology. Up to the end of the twentieth century, engineers put software into hardware. From the beginning of the twenty-first century, we are doing the opposite. We now design hardware using software (e.g., using CAD[10] systems and HITLS environments). Ultimately, we can 3D-print hardware from software (Fig. 2.1).

[10] Computer-aided design.

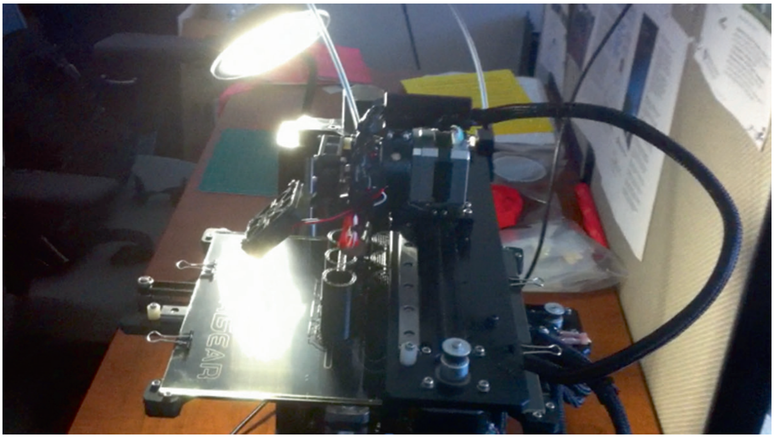

Fig. 2.1 A simple 3D printer that enables printing a CAD model

Consequently, this new information technology approach enables more holistic design and human-systems integration very early during the design process. However, resulting upstream tests rely on software prototypes that require tangibility. The notion of tangibility currently shifts from the previous inside-out philosophy to the outside-in philosophy. In the inside-out philosophy, the HSI burden was put on the end users. In the outside-in philosophy, the HSI burden is now put on the designers. In other words, designers need to understand the various cognitive functions involved in the use of the systems they are designing.

We will develop the cognitive function aspect in the next chapter. However, at this point, it is important to mention that some cognitive functions involved in the use of a system being designed can be deliberately defined by design, but there are other cognitive functions emerging from the use of the system that cannot be anticipated without human-in-the-loop simulations and sometimes real-world operations. Early discovery of emerging cognitive functions is another new possibility provided by information technology that supports HCD. This process of emerging cognitive function discovery is even more crucial in the design of large complex systems (e.g., design of a hospital). This is why complexity science has become crucial support in the design of such large systems.

Evolution, Revolution, and Constant Changes

The HFE tradition often dictates continuity of work practices. I remember the fear of automation when we delivered the first highly automated cockpits (Billings 1991); one of the arguments was the lack of continuity in work practices. This kind of automation was not an evolution; it was a revolution. Whenever work practice changes, it is however reasonable that people fear for their jobs. The main question is acceptability of new technology. We obviously can expect opposition, possible rejection, and, if accepted, possible surprises, when this technology is only technology centered. This is why HCD is necessary.

During the 1980s and 1990s, automation drawbacks emerged from several HFE studies, such as “ironies of automation” (Bainbridge 1983), “clumsy automation” (Wiener 1989), and “automation surprises” (Sarter et al. 1997). These studies did not consider the importance of maturity of technology, maturity of practice, and organizational maturity. These three concepts will be further developed in Chap. 7. Good design can be seen as a matter of function allocation. More specifically, functions cannot be correctly allocated among humans and machines without a thorough identification of **emerging cognitive functions**. In other words, we need to observe **user's activity** and not being limited to prescribed tasks. Only users facing new technology will be able to make emerge cognitive functions that were not anticipated by designers.

Automation surprises happen when a tangible interactive system is not mature enough and when people become too complacent to work with it.

Sometimes socio-technical evolution transforms into a revolution. This was the case of the revolution of the fly-by-wire that led to what we call today “interactive cockpits.” The accumulation of software-based systems in aircraft transformed pilots into aircraft system managers. Pilots' jobs now require them to know not only how to fly an airplane but also aircraft systems and how to manage them. It is clear that this is more than an evolution. In addition to knowing about mechanical systems of an airplane, pilots now need to know about systems controlling them (i.e., this requires more cognitive capacities in addition to flying skills).

This distinction between evolution and revolution has now to be considered as a dynamic process. Technology is changing faster than before. This is due to the extensive use of software both in the design of systems and in the systems themselves. Information technology increases connectivity at a very large scale, transforming socio-technical systems into biological-like systems. By analogy to Nobel Prize Jacques Monod's thesis in molecular biology, the constant changes that we experience today can be seen as **chance and necessity** (Monod 1970). Current information technology (IT) happened as a chance. It is also a necessity in many cases. For example, IT is both a chance and a necessity in air traffic management. Increasing connectivity among aircraft to handle high density and airport capacity progressively appears to be a necessity. However, in this case, IT also brings new problems such as cybersecurity. This is why the evolution of related socio-technical systems brings us into a new era, like *Homo sapiens* emerged as a distinct species of hominids.

Summarizing, for the last 60 years, HCD socio-technical evolution can be decomposed into three phases (Fig. 2.2):

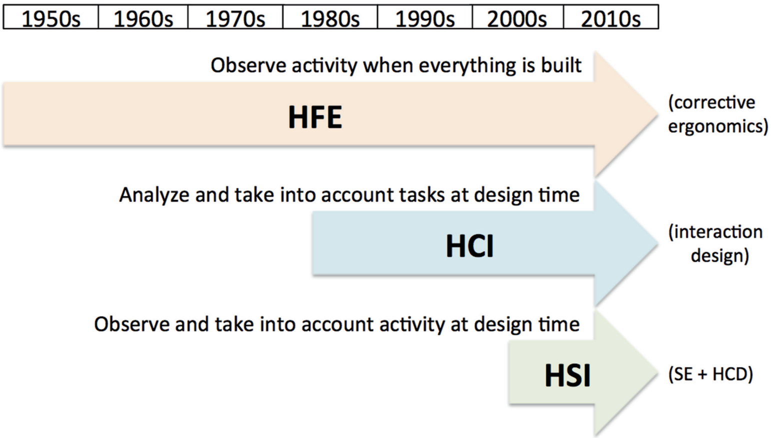

Fig. 2.2 Human-centered design evolution

- Human factors and ergonomics (HFE) that was developed after the World War II to correct engineering productions and had generated the concepts of human-machine interfaces or user interfaces.
- Human-computer interaction (HCI) that started to be developed during the 1980s to better understand and master interaction with computers; it contributed to shift from corrective ergonomics to interaction design.
- Human-systems integration (HSI) that emerged from the need of taking into account human factors in systems engineering (SE); SE and HCD combined incrementally lead to HSI.

In other words, the TOP (technology, organization, and people) model shifted from interface to interaction to integration. It also could be seen as a shift from engineering to information technology to systems, where systems include people and complex interactive artifacts. Finally, another interpretation is the shift from mechanical machines to computers to tangible interactive systems.

The Progressive Issue of Tangibility

Ray Kurzweil claims that information technologies grow exponentially, but our brain grows linearly (Kurzweil 2013). They double their power (i.e., price performance, capacity, and bandwidth) every year. Kurzweil advocates that our societies will evolve toward "super-intelligence, and with it, humans/machines expanding into the Universe.[11]" He predicts that by 2029, we will be able to reverse engineer our brain and make machines that will be far more powerful than our brains. He also

[11] http://bigthink.com/the-nantucket-project/ray-kurzweil-the-six-epochs-of-technology-evolution

predicts that we will stop aging and dying. If this kind of prediction is true, how will we regulate our planetary population growth? Knowing that we have gone from two billion people in the beginning of the twentieth century to more than seven billion now and possibly ten billion in 2060. A logical answer would be to say that we will be obliged to explore the universe and find other planets to live on. Why not?

Even if I currently observe this technology exponential growth (Emmott 2013), I also observe lots of troubles on and around our planet that need to be fixed, global warming, freshwater issues, diseases due to starving, and other serious problems due to the growth of our worldwide population and the distribution of wealth. Technology and its benefits for a small number of people are growing. These are facts! But how can we continue to develop this technology for the sake of everyone on planet Earth? Are there alternative ways of thinking? Information technologies provide crucial means to solve Earth problems, but they need to be used properly for the right purposes. Taking a HCD approach, I came to the point that the most important issue is tangibility (i.e., we need to be able to grasp these information technologies for dedicated specific purposes as well as integrate them together for bigger endeavors). Now let's figure out in more details what TISs are and could be.

Tangible Things Suddenly Emerge from Failing Virtual Things

I had an interesting experience planning to go to a country where I went several times. The procedure about getting a visa was very straightforward. I usually sent my passport to a broker company that took care of it. The delegation process always went perfectly in the past. They always did their job extremely well. This time, starting the procedure on their Web site, I suddenly discovered that they could not handle the visa process for French citizens any longer. In other words, I had to go to the consulate of the targeted country and do the paper work in person. I had to go through the whole process myself. In particular, I had to fly to the consulate and spend 2 days there. I realized that the process that was virtual to me before became very tangible in terms of time, money, travel, and concentration spent on details that were not familiar to me. This story shows that our society developed services that removed perception, understanding, and action on things that are very tangible if you had to do them by yourself. When these virtual services fail, we need to handle lower grain tangible things.

This is the same thing for any automated processes that fail. Human operators have to handle lower grain tangible things, in the physical sense. Consequently, it becomes crucial that everybody understands what tangible interactive systems are about. In the past, this visa company, which I was talking about, was a black box for me. I did not know people in this company. This company was therefore tangible for me, in the figurative sense, because it was able to satisfy my visa needs. Although I ignored the way it was working internally, I trusted it. This story made me realize that our consumer society tends to take for granted virtual services until they fail (i.e., in my visa story, the failure was an exception that forced me to analyze, understand, and execute the various elementary tasks involved). Indeed, when these services become unavailable, we need to handle directly things that were usually handled for us in the background. Indeed, I considered the visa company as a very useful TIS until this exception came out. More generally, a TIS should stay tangible (i.e., easy to understand and operate) when it fails.

Summarizing, during the twentieth century and before, we had to directly handle tangible physical objects, but now other people and systems are doing them for us in the background, except when they fail. Consequently, citizens of the twenty-first century will need to have deeper knowledge of what is going on in the background of services they are using or have services handling exceptions. In other words, education should take into account this requirement seriously. At this point, tangibility becomes a matter of acquaintance between people and systems.

From Inside-Out to Outside-In Tangibility

A distinction will be made between inside-out tangibility based on automation (i.e., the product of integrating software into hardware) and outside-in tangibility based on tangible interactive systems (i.e., the product of shaping hardware from and around software). As already mentioned, layers of automation were added to physical things as a solution for users' safety, efficiency, and comfort. The inside-out approach to engineering was initially based on technology development and human factor care after full development. We then ended up investigating tangibility of automation and figured out that well-done automation introduced considerably better usability in nominal situations, but rigidity in unexpected situations. Tangibility of automation in nominal situations corresponds to invisibility of automation (i.e., human operators forget that the system is automated and enjoy its use). Tangibility of automation in off-nominal situations is another story. It deals with new types of situation awareness, decision-making, and action taking. It requires that human operators know about how (automated) systems work, how to override them, and what they have done prior to shifting into the off-nominal situations. Human operators also need to know how to operate systems without automation (i.e., manual reversion). For example, pilots need to keep their flying skills on the most advanced highly automated aircraft. Therefore, inside-out tangibility of automated systems is context sensitive. An automated system will be said **inside-out tangible** when automation is invisible to its user in nominal situation and self-explanatory in off-nominal situations. Of course, complexity of an automated system will define the level of training of its users, especially in off-nominal situations.

Technology-centered engineering is means driven. It leads to automation and inside-out tangibility.

The outside-in approach characterizes HCD. It promotes human-systems integration from the beginning of the design process by using M&S capabilities, creativity and design thinking, as well as incremental formative evaluations. Basic components of the outside-in approach to design and engineering are TISs. TISs should be tested individually as well as integrated in their environments. HCD includes scenario-based design (i.e., scenarios need to be defined and developed to support analysis, design, and evaluations) and evaluation/validation principles and criteria. Scenarios are typically both declarative scenarios (i.e., systems configurations) and procedural scenarios (i.e., scripts and chronologies of use cases). Two categories of scenarios can also be distinguished: nominal scenarios and off-nominal scenarios. HITLS are used to test systems being developed based on scenarios and evaluation criteria. HCD is an incremental process that progresses with TISs' maturity. Chapter 7 is devoted to maturity of TISs. Therefore, outside-in tangibility of human-centered systems is also **context sensitive**, but, unlike inside-out tangibility, context sensitivity is tested from the very beginning of the design process. A human-centered system will be said **outside-in tangible** when its TISs are well integrated both among them and with their users. Complexity of a human-centered system will be investigated during the design process, and along the life cycle of systems being developed and operated. In other words, M&S will be used to support both creativity (i.e., TIS synthesis and integration) and rationalization (i.e., comprehension and validation of human-systems integration). We can expect a progressive standardization of TISs.

Human-centered design is purpose driven. It leads to TISs and outside-in tangibility.

Summarizing: What Can We Say on Tangibility at This Point?

Tangibility can be understood in the physical sense as well as in the figurative sense. In the physical sense, systems that we create and build should be graspable and well integrated in their physical environment. In the figurative sense, we often say, "I don't buy this idea; it is not tangible!" What does the term "tangible" means in the case of an idea? It means that you can or cannot grasp the concept behind the argument. It means that you can or cannot accept this idea using your own frame of reference.

Even if we developed great human-computer interaction capabilities based on cognitive engineering principles, cognitive factors are not the only human factors that need to be taken into account. Sensory-motoric factors are still there and require much attention. Our hands are effectors and also sensors. Gestures are incredible means for interaction. Considering our five senses, HCI mostly focused on central vision, providing computer screens where almost the whole underlying interaction is managed. Interaction devices, such as the mouse, enabled quick interaction with

Fig. 2.3 3D printed a handheld rocket pointing device

these screens. We are manipulating virtual objects on screens. The role of our hand (i.e., only one hand) has been standardized and reduced to specific horizontal movements and clicks. The use of TISs will bring different directions for interactivity. For example, we recently developed an interaction device that enable to quickly and purposefully display camera views around a rocket during the countdown period before a launch. We 3D printed a handheld mock-up of the rocket, equipped it with sensitive sensors, and tested it with a few cameras located around the actual rocket (Fig. 2.3). This TIS enables its user to point where he or she wants to see details on the rocket surface, and the system automatically provides the desired view from the closest camera. Experimentations showed that this is a very powerful and meaningful tangible interactive system.

Interaction with the physical environment using computers requires meaningful artifacts. For example, if we want to improve the quality of life in a large city such as Paris, transportation system integration is a real issue. How can we study such integration to better design new systems? First, we need to know about people's mobility in such a city. Is traveling more than 2 h in the morning tangible? The first answer that comes to mind is no! However, when you look carefully at what Parisians do, you immediately realize that they have adapted to the city system by necessity, either for money reasons or because they want to live in a specific area of the town and their work requirements bring them elsewhere. Information technology can propose new ways of working such as teleworking from home and working in the transportation system itself. This requires new technologies and most importantly infrastructures that are capable of handling mobility and work at the same time. On this very brief example, you can see that nothing is possible without creativity and an interdisciplinary approach.

Tangibility has also to be taken at the conceptual level. We communicate using advanced interaction media much more than a few years ago. For example, we use Skype, VSee, or Google Hangouts to communicate among friends but also at work.

In other words, we use virtual media to perform human-human interactions. For the last two decades, the computer-supported cooperative work (CSCW) community designed and developed technology to this end. Within this framework, what do we mean by tangible interaction? Sometimes it is difficult to perceive tangibility of ideas being discussed because we do not grasp clear physical attitudes of people remotely located. This is a physical (or physiological) tangibility issue. On the other hand, people can add text to oral interactions using chat capabilities. This increases credibility of what is being said, just because written arguments can be kept and further used (in the legal sense). This is a figurative tangibility issue.

Summarizing, tangibility is intimately related to physical sensing capabilities, personal desires, life management, infrastructure availability, human attitudes, and conceptual credibility. We often talk about tangible evidence. Let's provide a list of synonyms for tangible: current, authorized, assured, authentic, certain, concrete, consistent, demonstrated, effective, established, exact, existent, founded, unquestionable, undisputed, incontestable, indubitable, evident, concrete, objective, palpable, real, serious, solid, truthful, and true.

References

Bainbridge L (1983) Ironies of automation. Automatica 19(6):775–779. International Federation of Automatic Control, Pergamon Press

Billings CE (1991) Human-centered aircraft automation: a concept and guidelines, NASA technical memorandum 103885. NASA-Ames Research Center, Moffett Field

Bolter JD, Gromala D (2008) Windows and mirrors: interaction design, digital art, and the myth of transparency. MIT Press, Cambridge, MA. ISBN 0-262-02545-0

Boy GA (2013) Orchestrating human-centered design. Springer, London. ISBN 978-1-4471-4338-3

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, UK. ISBN 978-3-319-02811-8

Cannon WB (1932) The wisdom of the body. Reprinted by W.W. Norton & Company, New York; Rev. and Enl. Ed edn (17 Apr 1963). ISBN-13: 978-0393002058

Card S, Moran TP, Newell A (1983) The psychology of human computer interaction. Lawrence Erlbaum Associates, Hillsdale. ISBN 0-89859-859-1

Emmott S (2013) Ten billion. Vintage Books, New York. ISBN: 978-0-345-80647-5

Grudin J (2006) Turing maturing: the separation of artificial intelligence and human–computer interaction. Interactions 13(5):54–57

Ishii I, Ullmer B (1997) Tangible bits: towards seamless interfaces between people, bits and atoms. Proceedings of CHI'97, 22–27 Mar. ACM Digital Library

Kurzweil R (2013) How to create a mind: the secret of human thought revealed. Penguin Books, New York. ISBN 978-0143124047

Monod J (1970) Chance and necessity: essay on the natural philosophy of modern biology (in French: Le Hasard et la Nécessité: Essai sur la Philosophie Naturelle de la Biologie Moderne). Seuil. ISBN-13: 978-2020028127

Norman DA, Draper SW (eds) (1986) User centered system design: new perspectives on human-computer interaction. Lawrence Erlbaum Associates, Pub, Hillsdale

Rajkumar R, Lee I, Sha L, Stankovic J (2010) Cyber-physical systems: the next computing revolution. Design Automation Conference 2010, Anaheim, California, USA. ACM Digital Library.
Rogers Y, Sharp H, Preece J (2011) Interaction design: beyond human – computer interaction, 3rd edn. Wiley, Chichester, U.K. ISBN-13: 978-0470665763
Rosenblueth A, Wiener N (1945) The role of models in science. Philos Sci 2(4):316–321
Sarter NB, Woods DD, Billings CE (1997) Automation surprises. In: Salvendy G (ed) Handbook of human factors & ergonomics, 2nd edn. Wiley, New York
Sheridan TB (1984) Supervisory control of remote manipulators, vehicles, and dynamic processes. In: Rouse WB (ed) Advances in man-machine research. JAI Press, New York, pp 49–137
Sutherland J (2014) Scrum: the art of doing twice the work in half the time. Crown Business, New York. ISBN-13: 978-0385346450
Wiener EL (1989) Human factors of advanced technology ("Glass Cockpit") transport aircraft. NASA Contractor Report 177528. NASA Ames Research Center, Moffett Field, CA
Winograd T, Flores F (1987) Understanding computers and cognition: a new foundation for design. Addison-Wesley Reading, MA, USA. ISBN-13: 978-0201112979

Springer

http://www.springer.com/978-3-319-30269-0

Tangible Interactive Systems
Grasping the Real World with Computers
Boy, G.
2016, XII, 212 p. 61 illus., 38 illus. in color., Hardcover
ISBN: 978-3-319-30269-0