

## Human-centered design of complex systems: An experience-based approach

Guy André Boy

*School of Human Centered Design, Innovation and Art, Florida Institute of Technology, USA*

---

### Abstract

This position paper presents a new approach based on my experience in the evolution of human-centered design (HCD) during four decades, and how it has struggled to become a discipline in its own right in complex socio-technical systems' creation, development and operations. The 20th century saw tremendous industrial developments based on tangible materials that were transformed and assembled to make washing machines, cars, aircraft and power plants; during its last three decades, electronics and software were incrementally added to hardware machines. Operationalization issues moved from hardware to software, making automation and user interfaces central issues. From the beginning of the 21st century, we began to do the exact opposite! Currently, we typically start a project by designing and developing technology on computers, using software only, which is later transformed into hardware (and software). I denote this shift, the 'socio-technical inversion'. Operationalization issues are moving from software to hardware, making tangibility a central issue. Three useful conceptual models are presented: the SFAC (Structure/Function versus Abstract/Concrete) model; the NAIR (Natural/Artificial versus Cognitive/Physical) model; and the AUTOS (Artifact, User, Task, Organization and Situation) pyramid. Concepts developed in this article are based on the rationalization of a long experience in the aerospace domain.

**Key words:** human-centered design, human–systems integration, human activity, maturity, human–computer interaction

---

Received 8 September 2016

Revised 6 April 2017

Accepted 20 April 2017

#### Corresponding author

G. A. Boy  
gboy@fit.edu

Published by Cambridge  
University Press  
© The Author(s) 2017  
Distributed as Open Access under  
a CC-BY 4.0 license  
(<http://creativecommons.org/licenses/by/4.0/>)

*Des. Sci.*, vol. 3, e8  
[journals.cambridge.org/dsj](http://journals.cambridge.org/dsj)  
DOI: 10.1017/dsj.2017.8

the **Design Society**  
a worldwide community



### 1. Introduction

When I was offered the opportunity to write this position paper in the Design Science journal, I tried to better understand what design science is about. I first thought that 'design' and 'science' could be considered as very different concepts leading to two very different kinds of practice and culture. Science is strongly based on rigorous demonstration and validation of initial claims. Design is about creativity and innovation, considered as a new synthesis of existing materials at various levels of integration. Consequently, putting design and science together, as complementary disciplines, contributes to combine creativity, demonstration and validation.

Design contributes to the production of artifacts that could be useful to and usable by people. An artifact is any conceptual or physical object that is made by people. Science contributes to the production of knowledge that enables explanation and/or prediction of facts and events. Consequently, I consider (at least in this article) design science as a discipline that contributes to the

production of design knowledge useful to and usable for, the production of artifacts.

People have created artifacts for a long time, to support and improve their lives. Engineers have managed to optimize these artifacts in the mathematical sense (Papalambros & Wilde 2000). For that matter, the design of an artifact involves decision-making and engineering design, commonly supported by development of mathematical models that represent artifact structure(s) and function(s). Extension of these models to include people using or interacting with this artifact is a real problem since most human mathematical models are far from being representative, and even less predictive. There are very useful local mathematical models of physiological organs (e.g., respiratory or vascular functions models), but it is still impossible to develop global human mathematical models that take into account all 'significant' factors. In aeronautics, automatic control has entailed using human models that represent basic human pilot functions that enable landing a simulated aircraft in very restrictive conditions, for example.

More sophisticated *human operator models*, such as MESSAGE<sup>1</sup> (Boy & Tessier 1985) and MIDAS<sup>2</sup> (Corker & Smith 1993), were developed based on mechanistic architecture analogs such as Newell, Simon and Rasmussen's generic models (Newell & Simon 1972; Rasmussen 1983). Even if these human operator models were extremely complex (sometimes too complex to understand what they were doing!), they were far away from what real people could do. It is often better to use conceptual human models to understand and observe activity produced by real people in human-in-the-loop simulations (HITLS). This evolution from narrow human mathematical and logical models to conceptual human models requires understanding of human-centered design (HCD) history.

HCD has been described in several ways. It emerged as a reaction to the rigid world of corporate design and engineering (i.e., current systems engineering), which dictates that engineering leads to design and development, and people would be considered when technology would be developed by creating user interfaces and operational documentation.

In 2005, the Hasso Plattner Institute of Design (known as the d.school) was founded at Stanford University after the name of its major donator, SAP co-founder Hasso Plattner, based on the Design Thinking concept developed by David Kelley, Larry Leifer and Terry Winograd (Weinberg 2012). Design thinking takes 'context' into account (i.e., people's requirements, technological possibilities and economic viability) (Brown 2008). Design thinking brings flexibility that contrasts with analytical thinking rigidity (Plattner *et al.* 2016). In addition, design thinking incorporates creativity to conventional STEM<sup>3</sup> approaches, promoting a culture of innovation where HCD defines new STEAM<sup>4</sup> approaches. Finally, design thinking deals with change management, and therefore organization design.

HCD finds its roots in human-computer interaction (HCI), which takes into account human factors in computing systems and has also become a design discipline. Donald Norman is certainly one of the best promoters of HCD, where

<sup>1</sup> Crew Model of Crew and Aircraft Sub-Systems for Equipment Management (in French, Modèle d'Équipage et des Sous-Systèmes Avion pour la Gestion des Equipements).

<sup>2</sup> Man-Machine Integrated Design and Analysis System.

<sup>3</sup> Science, Technology, Engineering and Mathematics.

<sup>4</sup> Science, Technology, Engineering, Arts and Mathematics.



**Figure 1.** The TOP model (Boy 2013).

he recognized the need for observing activity,<sup>5</sup> making a difference between logic and usage. This leads to the concept of user experience (Edwards & Kasik 1974; Norman 1988). HCD encapsulates what Norman (1986) calls user-centered systems design (UCSD). The term ‘user’ may be misleading for two reasons. First, it lets people think about end users and not necessarily certifiers, maintainers and trainers, for example. Second, people who deal with a system have more characteristics than being users; they are people!

HCD of complex systems considers these philosophical distinctions for the concurrent creation and development of artifacts (concepts and/or technology), together with people and organizations that relates to them. Technology, organizations and people’s activities are co-designed and studied using different kinds of scientific methods. We now talk about the TOP model in HCD (Figure 1).

For example, NASA Human Systems Integration practitioner’s guide provides a very clear and explicit definition of HCD in the space domain (Rochlis Zumbado 2015):

- (a) concepts of operations and scenario development;
- (b) task analyses;
- (c) function allocation between humans and systems;
- (d) allocation of roles and responsibilities among humans;
- (e) iterative conceptual design and prototyping;
- (f) empirical testing, e.g., human-in-the-loop testing with representative population, or model-based assessment of human–system performance;
- (g) *in situ* monitoring of human–system performance during flight.

After providing a definition of complex systems, this article will provide an analysis of the shift from 20th century’s technology-centered engineering leading to automation and maturity issues (i.e., going from hardware to software) to 21st century’s HCD leading to tangibility and organizational issues (i.e., going from software to hardware). I denote this shift, the ‘socio-technical inversion’. This will enable us to show that we now not only need to set prescribed tasks but also observe human activity. In addition, I will provide my experience-based contribution to HCD fundamentals that support complex systems design. HCD will be presented as interdisciplinary teamwork and a process that enables discovering generic concepts from observation.

<sup>5</sup> [http://www.jnd.org/dn.mss/logic\\_versus\\_usage\\_the\\_case\\_for\\_activity-centered\\_design.html](http://www.jnd.org/dn.mss/logic_versus_usage_the_case_for_activity-centered_design.html).

## 2. What is a complex system?

The following properties typically characterize a complex system:

- (1) a large number of components and interconnections among these components;
- (2) many people involved in its life cycle that includes design, development, manufacturing, operations, maintenance and decommissioning;
- (3) emergent properties and behaviors not included in the components;
- (4) complex adaptive mechanisms and behaviors – this can be called adaptability;
- (5) nonlinearities and possible chaos – this can be called unpredictability.

Examples of complex systems are aircraft, industrial power plants and large defense systems. They typically involve many people to design, manufacture, use, repair and dismantle them. In contrast, a simple system can be defined by the following properties:

- (1) a small number of components and interconnections;
- (2) behaviors directly related to components;
- (3) no or very simple adaptive mechanisms and behaviors;
- (4) linear or slightly linear responses to inputs.

Examples of simple systems are tables, cars and electronic watches. They do not require involvement of many people, except in the case of mass production.

From a design perspective, we distinguish and combine structural and functional complexity. The former is related to system structure,<sup>6</sup> sub-structures and so on. The latter is related to system function, sub-functions and so on. For example, the life cycle of an aircraft involves complex processes that deal with a large number of complex systems. Therefore, several sub-systems need to be articulated structurally and functionally. Consequently, several articulated backgrounds are required to design and manufacture a complex system, such as an aircraft. There is no space for improvisation. Whether they are designers, manufacturers or human operators, people who deal with complex systems need appropriate levels of familiarity with them and the environment they induce. This is what human–systems complexity analysis is about.

Some contributors talk about ‘complex socio-technical systems’ to emphasize people and technology (Grudin 1994; Carayon 2006; Baxter & Sommerville 2010; Norman & Stappers 2016). *I will keep the term, ‘complex systems’, considering that the corresponding concept necessarily include people.*

Finally, HCD of complex systems is necessarily interdisciplinary, since nobody can provide all possible contributions to the design of such systems, but a well-formed team can. Therefore, collaborative work is an important part of HCD (Poltrone & Grudin 2003). These concepts will be further developed in the article, using my concrete aerospace experience to illustrate them and make them more tangible.

<sup>6</sup> Structure and function concepts are taken from biology (e.g., the structure of the human lung is essential for its function). I extend these concepts to design and engineering (e.g., the structure of a spacecraft is essential for its function).

### 3. The socio-technical inversion

#### 3.1. 20th century automation and maturity issues

During the 20th century, mechanical engineering was the major discipline in engineering. This great technological advancement allowed engineers to make washing machines, cars, aircraft and nuclear power plants, for example. Engineers assembled tangible objects to make complicated machines. A negative impact of this rapid evolution was discovered; the use of these machines was often discovered too late to make appropriate socio-technical corrections. Consequently, Human Factors and Ergonomics (HFE) specialists' contributions were always possible only at the end of the development process, and therefore they were conflicting with engineers. In other words, HFE contributions were often not effective because operability (i.e., usability and usefulness of systems being designed and developed) was tested too late to provide results that could be integrated. Drastic modifications were almost impossible without spending big amounts of money. Corrective ergonomics led to user interface design (Vicente 2002).

HCI brought *task analysis* a step further because it moved from manipulation of physical objects to interaction with software-based objects, available on computer screens. However, the scope of HCI stayed limited to user interface design.<sup>7</sup> For example, HCI techniques developed for office automation were transferred to aeronautics, transforming the conventional aircraft cockpit into what we call today 'interactive'<sup>8</sup> cockpits. Pilots now interact with onboard computers using a pointing device, but not directly with aircraft mechanical surfaces like in the past.

HFE tradition often fears discontinuity of work practices. I remember the fear of automation when the first highly automated cockpits were delivered during the late nineteen eighties; one of the HFE arguments was lack of continuity in work practices. This kind of automation was not a casual evolution of pilots' work; it effectively led to a socio-technical revolution, which required that pilots knew not only how to fly an airplane but also how to manage onboard systems. At that time, HFE specialists did not have the right philosophy and practice to evaluate such a change. We needed to develop new concepts that led to a new discipline, cognitive engineering (Norman 1986). The challenge was to better understand the socio-cognitive consequences coming from the shift *from control to management*, which emerged from the incremental addition of software into hardware. This phenomenon has been called automation.<sup>9</sup> For example, pilots needed to know about and how to manage digital systems (i.e., this requires more cognitive capacities in addition to flying skills). The art of conventional flying incrementally became a matter of onboard system management.

During the eighties and nineties, 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

<sup>7</sup> In contrast with current HCD approach that is inherently holistic at the (complex) system level (i.e., using the TOP model).

<sup>8</sup> The term 'interactive' should be taken in the HCI sense.

<sup>9</sup> It should be noted at this point that autopilots were introduced and used on commercial aircraft in the early nineteen thirties (e.g., the Boeing 247 aircraft). They were not based on software, but analogic technology. During the early eighties, digital technology introduced a huge amount of possibilities, and automation never stopped to evolve toward avionics software engineering.

studies did not consider the importance of *technology maturity* and *maturity of practice*. Automation can be modeled as cognitive function transfer from people to systems (Boy 1998).<sup>10</sup> If automation considerably reduced casual people's burdens, it also caused problems such as complacency, which is an emerging cognitive function (i.e., not predictable at design time, but at operations time).

Good design can be seen as optimal function allocation. Human and system function allocation cannot be limited to *a priori* optimal assignment of prescribed tasks to humans and machines, in Fitts's sense (Fitts 1951); it should be extended to the identification of *emerging*<sup>11</sup> functions that are only observable at use time. This is the reason we need to observe what people really do.

Automation was designed as a single agent entity (i.e., in isolation from the world around) and required considering technology and practice maturities.

### 3.2. 21st century tangibility and organizational issues

Today, we develop an entire aircraft on computers from inception of design to finished product. Therefore, we can test its operability from the very beginning, and along its life cycle using HITLS and an agile<sup>12</sup> approach using virtual prototypes. Consequently, operability of complex systems can then be tested during design. This is the reason why HCD has become a discipline in its own right. HCD enables us to better understand HSI during the design process and then have an impact on requirements before complex systems are fully developed. However, even if these environments are very close to the real world, their *tangibility* must be questioned, and most importantly validated.

What does the tangibility concept mean exactly? It has two meanings. First, *physical tangibility* is the property of an object that is physically graspable (i.e., you can touch it, hold it, sense it and so on). Second, *figurative tangibility* is the property of a concept that is cognitively graspable (i.e., you can understand it, appropriate it, feel it and so on). If I try to convince you about something, you may tell me, 'what you are telling me is not tangible!' This means that you do not believe me; you cannot grasp the concept I am trying to provide. We also may say that you do not have the right mental model to understand it, or I do not have enough empathy to deliver the message correctly.

Tangibility is about physical and cognitive situation awareness. For example, we first developed an Onboard Context-Sensitive Information System (OCSIS) for airline pilots on a tablet PC (Tan & Boy 2016). Physical tangibility considerations led to a better understanding of whether OCSIS should be fixed-based in the cockpit or hand-held. Other considerations led to the choice of displays of weather visualization going from vertical cylinders to more realistic cloud representations

<sup>10</sup> Cognitive function analysis was developed to support the analysis, design, and evaluation of interactive systems, based on a socio-cognitive model that involves the concept of cognitive function (Boy 1998). A cognitive function is typically represented by its role, context of validity and mandatory resources to perform tasks attached to it.

<sup>11</sup> The concept of 'emergence' needs to be understood in the complexity science sense. An emergent property of a complex system emerges from interactions among its components (and sub-components) that do not exhibit such a property.

<sup>12</sup> 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).

(figurative tangibility). A set of pilots gave their opinions on various kinds of OCSIS tablet configurations. It is interesting to note that the pilots always naturally used the term ‘tangible’ to express their opinions.

Therefore, tangibility metrics should be developed to improve the assessment of complex systems operability. This is where subject matter experts and experienced people enter into play. We absolutely need such people in HCD to help assess HSI tangibility. For example, very realistic commercial aircraft cockpits, professional pilots and realistic scenarios are mandatory to incrementally assess tangibility. OCSIS was tested from the early stages of the design process using HITLS, by recording what pilots were doing using it and analyzing produced activity. Such formative evaluations lead to system modifications and improvements.

While the 21st century shift from software to hardware is not necessarily obvious, it is the next dilemma we must address, especially now that we can 3D print virtual systems and transform them into physical systems. We will denote resulting systems, Tangible Interactive Systems (TISs) (Boy 2016). TISs are strongly based on the multi-agent concept, unlike 20th century automation that was usually based on the single agent concept. This is why *TISs cannot be taken into account without a human–systems organizational approach*. More generally, co-evolution of people’s activities and technology necessarily led to a tangible organizational evolution.

A shift from the old army pyramidal model to the *Orchestra model*<sup>13</sup> is currently emerging (see the Orchestra model in Boy 2013). For example, technology and emergent practices have led people to change ways of communicating among each other. The army model induced vertical communication, mostly descendent. Transversal communication (e.g., using telephone, email and the Web) contributed to the emergence of the orchestra model. This functional evolution is now changing organizations themselves (i.e., structures). For example, smart phones and the Internet have contributed to change in both industrial and everyday life organizations.

Having this organizational model in mind, it is now crucial to use it in HCD. In the next section, we will examine how structures and functions determine each other, and how HSI is a matter of cognitive intentionality and physical reactivity. Note that even if ‘structure’ is often denoted as ‘form’ in design and architecture, we will keep the term ‘structure’ within the scope of this article.

Tangibility emerges from a multi-agent system (i.e., taking into account the interconnected world around) and requires investigating organization maturity.

## Summarizing

Automation was a 20th century critical issue (moving from hardware to software); tangibility is a 21st century crucial issue (moving from software to hardware).

<sup>13</sup> The *Orchestra model* provides a usable framework for human–systems integration (HSI). It requires definition of a common frame of reference (music theory), as well as jobs such as the ones of human-centered designers and systems architects (composers) who provide coordinated requirements (scores), highly competent socio-technical managers (conductors) and performers (musicians), and well-identified end users and involved stakeholders (audience).

## 4. The task–activity distinction to better understand HCD evolution

At this point, let us clarify the relationship between the task–activity distinction and socio-technical disciplines (i.e., HFE, human–computer interaction, human–systems integration). What do task and activity mean?

*A task is a prescription to people (e.g., human operators or users).* Task analysis is a very popular practice in human factors engineering (Wickens *et al.* 2003). It is often performed *a priori*, as it is done in technology-centered engineering.

*An activity is what people effectively do.* Even though Paul Fitts's work stressed the importance of observing activity rather than specifying tasks, he and his colleagues limited their observation to human errors; that is the negative part of activity only (Fitts & Jones 1947). Activity theories were first developed by the Russian School of Psychology (Leont'ev 1981; Kaptelinin 1995) and French School of Ergonomics (Leplat 1976; Leplat & Hoc 1983). The concept of activity also refers to distributed cognition (Hutchins 1995). Finally, activity is also defined in my model of cognitive function, which connects task and activity (Boy 1998), and is strongly inspired from these models, also anchored in situated cognition approaches (Suchman 1987) and embodied cognition (Varela *et al.* 1999). We can also say that compilation of people's activities provides people's experience (Norman 1988).

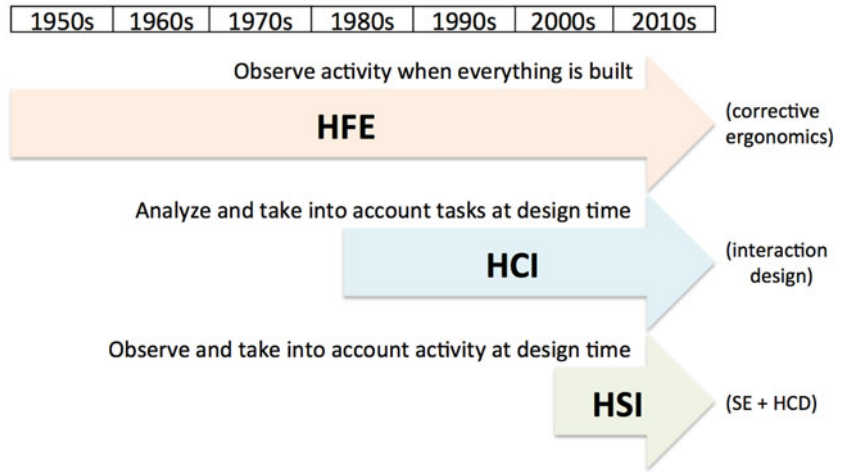
For the last sixty years, socio-technical evolution can be decomposed into three phases that made three communities<sup>14</sup> emerge (Figure 2):

- *Human Factors and Ergonomics* (HFE)<sup>15</sup> was developed after the second world war to correct engineering production, and generated the concepts of human–machine interfaces or user interfaces, and operational procedures; activity-based evaluation could not be holistically performed before products were finished or almost finished, which enormously handicapped possibilities of re-design. Sometimes, activity analyses were carried out prior to designing a new product, based on existing technology and practice; however, this HFE approach forced continuity, reduced risk taking, and most of the time prevented disruptive innovation.
- *Human–Computer Interaction* (HCI) started to be developed during the 1980s to better understand and master human interaction with computers;

<sup>14</sup> The author is qualified to talk about these three communities. He is still the Chair of the Aerospace Technical Committee of the International Ergonomics Association (IEA), which encapsulates most HFE societies around the world. From 1995 to 1999, he was the Executive Vice-Chair of the Association for Computing Machinery (ACM) Special Interest Group on Computer Human Interaction (SIGCHI), and Senior Member of the ACM. He is currently Chair of the Human Systems Integration Working Group of the International Council on Systems Engineering (INCOSE).

<sup>15</sup> There was a controversy between two supposedly different disciplines: human factors (mostly developed in North America) focusing on people using machines; and ergonomics (mostly developed in Europe) focusing on the adaptation of machines to people. A few years ago, a group of research scientists proposed a position on development of the already established discipline known as Human Factors and Ergonomics (HFE) (Dul *et al.* 2012), coordinated at the world level by the International Ergonomics Association (IEA). They wrote, 'HFE has great potential to contribute to the design of all kinds of systems with people (work systems, product/service systems), but faces challenges in the readiness of its market and in the supply of high-quality applications. HFE is a unique combination of three fundamental characteristics: (1) it takes a systems approach (2) it is design-driven and (3) it focuses on two closely related outcomes: performance and well-being'. My experience has shown me that HFE remains, for a very large part, a system evaluation discipline.





**Figure 2.** Human-centered design evolution.

it contributed to the shift from corrective ergonomics to interaction design mainly based on task analysis. Activity-based analysis started to be introduced within the HCI community by people who understood phenomenology (Winograd & Flores 1987) and activity theory (Kaptelinin & Nardi 2006).

- *HSI* emerged from the need to officially consider human possibilities and necessities as variables in systems engineering (SE); incrementally combined, SE and HCD lead to HSI, to take care of systems during their whole life cycle (Boy & Narkevicius 2013). HSI involves more than human factors evaluations or task analyses. More importantly, it *involves activity analysis at design time using virtual prototyping and HITLS* (e.g., we can model and simulate an entire aircraft, fly it as a computing game, and observe a pilot’s activity). It also involves creativity, system thinking, risk taking, prototype development using agile approaches, complexity analyses, organizational design and management, as well as HSI architecture knowledge and skills.

I do not want to create polemical discussions on labels, but say that there is a community around HFE, stable since World War II, with its own conferences and journals. The HCI community is clearly stable since the early nineteen eighties, with its own conferences and journals. Today, a new paradigm is emerging around HCD, HSI<sup>16</sup> and more generally, humanization of systems engineering. I know that HCD and HSI are still unstable terms, but we need to find a denotation for what we experience in the design and engineering world today.

<sup>16</sup> NASA defines HSI as, Human Systems Integration: An interdisciplinary and comprehensive management and technical process that focuses on the integration of human considerations into the system acquisition and development processes to enhance human-system design, reduce lifecycle ownership cost, and optimize total system performance. Human-system domain design activities associated with manpower, personnel, training, human factors engineering, safety, health, habitability, and survivability are considered concurrently and integrated with all other systems engineering design activities (Rochlis Zumbado 2015).

In this article, I choose the term *HCD* to denote a design discipline set up to support HSI architects, who define and refine prototypes that will be further developed as final products by engineers. When we talk about considering human factors at design time, we talk about HCD, which requires more vision, technology, knowledge and skills than what current HFE can offer.

## 5. HCD fundamentals : My experience-based view

Designing systems for people requires knowing what both systems and people are about. Let us present these models that provide appropriate concepts and the relationships among them.

### 5.1. The SFAC model

Designing an artifact is defining its structure and function. Each structure and function can be described in an abstract way and a concrete way. The *SFAC* model (Structure/Function versus Abstract/Concrete) provides double articulation (i.e., abstract and concrete) between artifact structure and function (Figure 3) as follows: declarative knowledge (i.e., abstract structures); procedural knowledge (i.e., abstract functions); static objects (i.e., concrete structures); and dynamic processes (i.e., concrete functions).

The abstract part is a rationalization of the system being designed (i.e., knowledge representation). This rationalization can be represented by a set of concepts related among each other by typed relationships. This kind of representation can be called ontology,<sup>17</sup> semantic network or concept map. It can take the form of a tree hierarchy in the simplest case, or a complex concept graph in most cases.

The 'declarative' and 'procedural' terms<sup>18</sup> respectively refer to knowing what and knowing how. They are used to describe human memory. Declarative memory includes facts and defines our own semantics of things. Procedural memory includes skills and procedures (i.e., how to do things). We can think declarative memory as an explicit network of concepts. Procedural memory could be thought as an implicit set of know-hows. Declarative memory and procedural memory are both in the cortex and involve learning. The former is typically stored in the temporal cortex of the brain. The latter is stored in the motor cortex.

At design time, the concrete part is commonly represented using computer-aided design (CAD) software, which enables the designer to generate 3D models of various components of the system being designed. These 3D models include static objects and dynamic processes that allow visualization of the way components being designed work and are integrated together. Later during the design and development process, these 3D models can be 3D printed, allowing for a more graspable appreciation of the components being built as well as their possible integration. Testing occurs at each step of the design process by considering

<sup>17</sup> In philosophy, ontology is the study of what there is, what exists. It is 'what the most general features and relations of these things are' (Hofweber 2011).

<sup>18</sup> These terms are used in computer programming that began by being thought and implemented in the procedural way. For example, languages such as Fortran were developed on the basis of subroutines, and Pascal used procedures. Subroutines and procedures enable programmers to develop procedural knowledge. Then artificial intelligence came and proposed declarative programming, such as Lisp and Prolog (i.e., defining objects, functions, predicates, and methods).

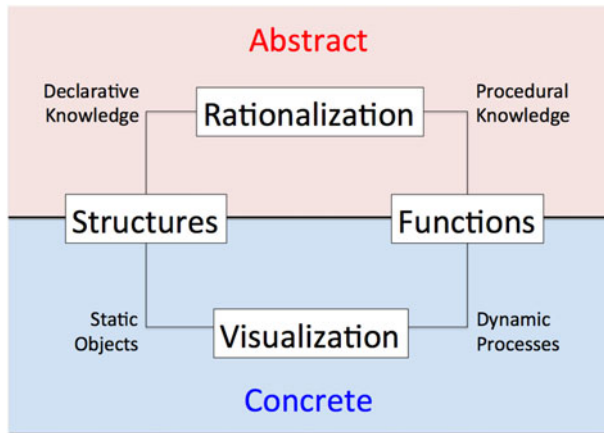


Figure 3. The SFAC Model (Boy 2016).

concrete parts together with their abstract counterparts (i.e., their rationalization, justifications and various relationships that exist among them).

The SFAC model is typically developed as a mediating space that design team members can share, collaboratively modify and validate. SFAC also enables the design team to support *documentation of the design process and its solutions* (Boy 1997). The concept of active design document (ADD), initially developed for traceability purposes, is useful for rationalization of innovative concepts and incremental formative evaluations (Boy 2005). The SFAC model was the basis of the SCORE system used to support a team, designing a light water nuclear reactor, in their collaborative work and project management (Boy *et al.* 2016).

## 5.2. The NAIR model

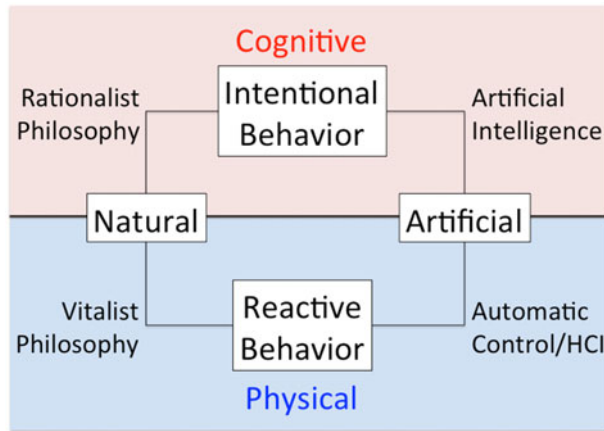
TISs cannot be studied, modeled, designed and developed if the distinction and complementarity of cognitive functions and physical functions is not well mastered. The (Natural/Artificial versus Cognitive/Physical) NAIR model is an attempt to rationalize this distinction for HCD (Figure 4).

Natural or artificial software-based systems have functions that are either cognitive or physical. Natural systems include biological systems of any kind, such as people, and physical systems (e.g., geologic or atmospheric phenomena). Artificial systems include information technology, such as aircraft flight management systems, Internet and mechanical systems such as old mechanical watches.

Situation awareness, decision-making and action taking are three essential cognitive functions of any human being. They produce an intentional behavior. Symmetrically, physical functions produce reactive behavior. Artificial intelligence tools and techniques can support intentional behavior (e.g., aircraft FMSs<sup>19</sup> use operations research, optimization techniques and knowledge based systems); control theories and human–computer interaction tools and techniques can support reactive behavior (e.g., aircraft TCASs<sup>20</sup> use radars,

<sup>19</sup> Flight Management Systems.

<sup>20</sup> Traffic alert and Collision Avoidance Systems.



**Figure 4.** Cognitive and physical functions: The NAIR Model (Boy 2016).

control mechanisms and voice outputs). On the natural side, intentional behavior can be supported by rationalist<sup>21</sup> philosophies (i.e., mainly related to the cortex, including reasoning, understanding and learning); reactive behavior can be supported by vitalist<sup>22</sup> philosophies (i.e., mainly related to the reptilian brain, including emotions, experience and skills).

### 5.3. The AUTOS pyramid

The Artifact, User, Task, Organization and Situation (AUTOS) pyramid extends the TOP model. It is a framework that helps rationalize HCD and engineering. It was extensively described in the introduction of the Handbook of Human–Machine Interaction (Boy 2011). The AUT triangle (Figure 5) describes three edges: task and activity analysis (U-T); information requirements and technological limitations (T-A); ergonomics and training (procedures) (T-U).

For example, artifacts may be aircraft or consumer electronics systems, devices and parts. Users may be novices, experienced personnel or experts, coming from and evolving in various cultures. They may be tired, stressed, making errors, old or young, as well as in very good shape and mood. Tasks vary from handling quality control, flight management, managing a passenger cabin, repairing, designing, supplying or managing a team or an organization. Each task involves one or several cognitive functions that related users must learn and use.

The organizational environment includes all team players, called ‘agents’, whether humans or artificial systems, interacting with the user who performs the task while using the artifact (Figure 6) (Boy 1998). It introduces three additional

<sup>21</sup> Rationalist philosophy is usually represented by René Descartes, Gottfried Wilhelm Leibniz and Baruch Spinoza (Benedict) Spinoza, who contributed to the development of mathematical methods during the Age of Reason (17th century), as well as Voltaire and Jean-Jacques Rousseau during the Age of Enlightenment (18th century).

<sup>22</sup> Henri Bergson promoted vitalism as human processes of immediate experience, free will and intuition over rationalism to better understand reality (Bergson 2001). Vitalism can be related to Friedrich Nietzsche’s will to power, based on the rejection of the distinction between organic and inorganic nature. The will to power can explain physical processes like eruption of a volcano. Nietzsche said that ‘life is merely a special case of the will to power’ (Nietzsche, edited by Kaufmann 1968).

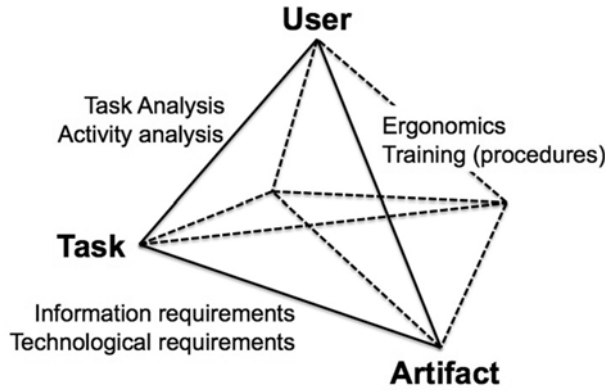


Figure 5. The AUT triangle.

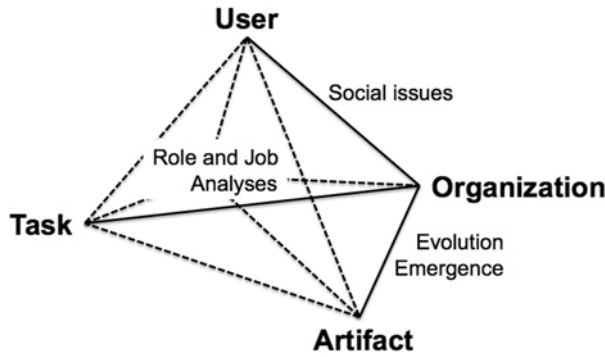


Figure 6. The AUTO tetrahedron.

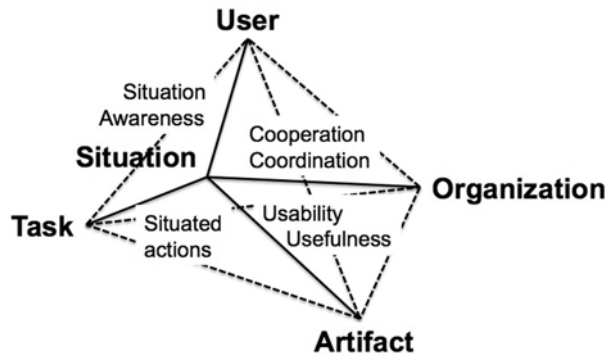


Figure 7. The AUTOS pyramid.

edges: social issues (U-O); role and job analyses (T-O); emergence and evolution (A-O).

The AUTOS framework (Figure 7) is an extension of the AUTO tetrahedron that introduces a new dimension, the ‘Situation’, which was implicitly included in the ‘Organizational environment’ (Boy 2011). The three new edges are:

usability/usefulness (A-S); situation awareness (U-S); situated actions (T-S); cooperation/coordination (O-S).

The AUTOS pyramid is useful support to human-centered designers in the analysis, design and evaluation of HSI, by considering human factors (i.e., user factors), systems factors (i.e., artifact factors) and interaction factors that combine task factors, organizational factors and situational factors.

## 6. HCD as interdisciplinary teamwork

When I started developing the HCD graduate program at FIT, I faced the difficulty of integrating an eclectic set of disciplines that are needed to make a good human-centered designer. First, I was convinced that such a human-centered designer should already have a background in engineering, science and/or architecture. Why? The reason is simple: he or she would need to develop prototypes (i.e., be acquainted with software and/or hardware). After a while, I realized that collaborative work was essential because everybody cannot have such engineering skills to concretely develop and test a great idea. A group of well-chosen people can, by participating in a well-orchestrated HCD team. Consequently, I developed and taught the following *six themes* that future human-centered designers had to learn, articulate and apply.

### 6.1. HCD contributing themes

*Cognitive engineering* was born in the early nineteen eighties (Norman 1986). Human-centered designers should know about it because they need to understand cognitive modeling, human errors and engagement, situation awareness and decision-making (non-exhaustive list). Cognitive engineering is about using cognitive science concepts and methods in design and engineering (Boy 2003; Boy & Pinet 2008).

*Advanced interaction media* is about human–computer interaction today. Human-centered designers should know about advanced techniques and tools for system control, data visualization, ubiquitous computing, socio-media and so on. Advanced interaction media requires using the latest information technology in HCD.

*Modeling and simulation* (M&S) is one of the strongest pillars of HCD. HCD could not exist before M&S became tangible (i.e., efficient, easy to use, realistic and comfortable). M&S enables HITLS, making activity observation and analysis possible. It enables user requirements development in complex systems design.

*Organization design and management* (ODM) is another pillar of HCD (refer to the TOP model and AUTOS pyramid). HCD is seldom possible in an organization that is not prepared and structured to welcome it. ODM provides knowledge and skills to further understand why culture, organizational structures, politics and personalities can influence HCD, and how HCD can change them.

*Complexity analysis* is crucial in the design of complex system. It is about analyzing effects of a large number of components (people and systems) and interconnections among these components, looking for emergent properties and behaviors not included in the components, understanding adaptability and unpredictability. Complexity science is not really taught at school, and HCD cannot be effectively done if human-centered designers do not know about it, at least at the level of first principles. Students need to know about context changes

and how to handle them. They also need to know about the effect of numbers (i.e., emerging effects of interactions among a large set of entities).

*Life-critical systems* (LCS) need to be categorized with respect to safety, efficiency and comfort. Human-centered designers should know about LCS properties, and make a distinction between internal and external complexity – this is usually related to technology maturity and maturity of practice (Boy 2013), as well as organizational maturity. Complex systems reveal their complexity when people interact with them. Very often the opposite of complexity is not only simplicity, but also *familiarity*.

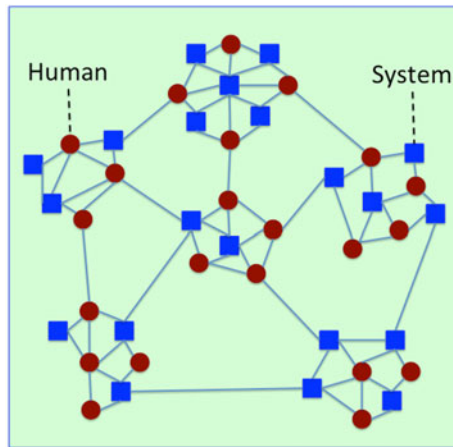
## 6.2. Participatory design

One single person often cannot perform interdisciplinary work. It is hard to be an expert in everything. This is the reason people need to work in teams to perform interdisciplinary work. *Cooperative work* is then an important philosophy and practice. HCD cannot be implemented without cooperative work. People need to participate for the whole team to succeed. This is what we usually call ‘participatory design’. Participatory design was developed intensely during the 1960s and 1970s in Scandinavian countries. Among several initiatives and work efforts, the Utopia project is a great example of co-design of technology, organizations and jobs based on hands-on experiences (Bødker *et al.* 1987).

Participatory design<sup>23</sup> requires collective situation awareness, empathy, and familiarity among design team members. *Collective situation awareness* is a matter of sharing purposes, current status of work in progress and a holistic view of the complex system being designed. Through collective situation awareness, participatory design should support intersubjectivity (i.e., design team members share the same meaning about what they are collectively designing). *Empathy* helps each design team member to understand and share the feelings of another. The more design team members become *familiar* with the complex system being designed (i.e., have a correct holistic view), the more they can articulate what they are devoted to do and what the other people, involved in other relevant components, do.

In groups, *creativity* emerges from the integration of various kinds of knowledge and skills toward satisfaction of goals and purposes. Teams, organizations and communities have their own properties. Teams are small, very fast, effective and highly collaborative. Organizations are large, very slow and very hierarchical. Communities could be small or big; they can be slow or fast depending on the domain; they can be highly collaborative. In all three types of groups, *leadership* and *followership* is required for each group member (i.e., if an expert leads the group today, he or she might not be the expert another day, and become a follower).

<sup>23</sup> Participatory design is a matter of knowledge and skills described in the previous paragraph, as well as domain knowledge. Coordination of the various disciplines involved depends on the type of group, whether it is a team (around ten people very much connected among each other), a structured organization and a community. *Teams* have existed for millions of years. *Organizations* have existed for several thousand years. *Communities* have existed for a few hundred years. Members of teams are usually collocated, as organization employees can be distributed in several locations, and communities distributed around the world. Team members share a common goal and are glued together through motivation. Organization employees are experts in a given domain and need to share common interests.



**Figure 8.** System of systems.

### 6.3. Orchestrating complex systems HCD: Team of teams

The 21st century is open, complex, dynamic and uncertain. We cannot design technology in a context-free framework any longer. More specifically, we need to situate industrial engineering (i.e., take into account context). *Context* is a matter of interaction among human and machine agents, and surrounding objects. Before delivering a new complex system, it is always better to anticipate possible emerging activities, properties and behaviors. Prototyping contributes to accelerate such anticipation. This context, which I am talking about, is perceived from the outside, but there is also context perceived from the inside (i.e., by each human agent<sup>24</sup> of the system). Each agent should, in many circumstances, know about current context of a complex system.

When human agents achieve more *autonomy*, the overall organization becomes more decentralized and more interconnected. Consequently, agents require more *coordination rules* and more explicit shared context. This is a matter of appropriate organizational model, as well as individual competence and empathy. I have already described the shift from the old army model (i.e., hierarchical, mostly descendent, information flow) to the orchestra model (i.e., transversal multi-directional information flow). Like musical instrument makers and composers would architecturally design new musical instruments that determine new kinds of symphonies or concertos, human-centered designers are architects of new technology that determines new kinds of activities.

For example, the design of a commercial aircraft is a huge enterprise that associates engineers (musical instrument makers) and pilots (composers). As HSI architects, human-centered designers should take into account engineers' and pilots' activities and jobs from the beginning of the design process to certification of the aircraft. Making a large aircraft, or more generally a complex system, requires several interdisciplinary teams working in concert. Consequently, we need to think in terms of team of teams (Leifer 2016), to match the concept of system of systems (Figure 8).

<sup>24</sup> At this point, I consider that we cannot develop good enough machine agents (technology) to perceive, process and act on context, even if some of them could handle very local contexts (e.g., diagnosing and recovering from specific system failures or human errors).



The Orchestra metaphor is then extended to *orchestra of orchestras* in a wide sense since we include all stakeholders from composition (i.e., HCD) to performance. Since this team of teams approach to HCD is decentralized and relies on autonomous agents, it requires strong individual competence and coordination. It also requires strong individual empathy and motivation.

### 7. Discovering generic concepts from observations during design

Aerospace activities encapsulate both procedure-following and problem-solving tasks and functions. Astronauts and ground personnel never stop switching from one to the other. In this section of the article, I will use the virtual camera concept, initially developed in the context of the NASA Lunar Electric Rover<sup>25</sup> (LER) for the exploration of the Moon (Boy *et al.* 2010; Boy & Platt 2013; Platt 2013; Platt *et al.* 2013; Platt & Boy 2014), describing its salient parts that enable presentation of generic HCD principles. Aerospace and more generally complex life-critical system domains involve expert and experienced human operators, which is not the case in public domains such as telephony and office automation.

One day, we were testing the LER prototype at Johnson Space Center in Houston, Texas. An astronaut was driving the rover on a Lunar-like hill, and I noticed that he was constantly asking people outside if he could go right, left, forward and backwards. This was because the hill was quite steep on one side and he did not want to fall down in the 'crater'. People outside were helping him as if a person is attempting to park a car and does not have enough visibility. I thought about this possible scenario when he will be on the Moon, and nobody will be present to guide his maneuvers. This is the reason I proposed development of a virtual camera system based on what we know (e.g., NASA already provided Moon data to make Google Moon). This kind of geographical data could be used to help astronauts navigate more safely by providing more situation awareness. However, what would happen when astronauts would go to areas where nothing is known? The virtual camera system could be connected to real cameras and other appropriate sensors that would provide images and space data, which could be fused with existing data. When this process is done incrementally, the Moon surface data can be incrementally updated, and used not only as a navigation system but also as an exploration support system.

The virtual camera project began as a problem-driven case. Participatory design, agile development and formative evaluations led to exploring a domain that was much broader than expected. We incrementally discovered emerging properties of the virtual camera concept, as well as inducing implemented technology and related user experience. The *generic virtual camera concept* became tangible because appropriate information technology was available, such as digital cameras, big data management, data fusion, data visualization and hand-held computing devices. Incrementally designing, developing and testing virtual camera applications for space exploration, using progressively refined prototypes, led us to elicit fundamental problems to be solved as well as other domains of application. We incrementally realized that the virtual camera was

<sup>25</sup> The Lunar Electric Rover was initially called Small Pressurized Rover and is called today Space Exploration Vehicle.

a generic concept and tool that supported this endeavor. Let us present two examples.

Almost 70% of delays in big airports are due to incorrect weather planning. When pilots are facing a large convective front for example, they need to execute a maneuver that may induce a delay in their approach and landing phase of flight. Current technology provides limited short-term account of weather situation. By extending the virtual camera concept, we are developing the Onboard Weather Situation Awareness System (OWSAS) that enables pilots to have 3D visualization of weather information in the cockpit together with their trajectory and the trajectories of other surrounding aircraft (Laurain *et al.* 2015; Boulnois & Boy 2016).

The virtual camera concept has also been extended to crisis management support for decision-makers. Let us imagine a catastrophe like the one that occurred in Fukushima, Japan, in 2011. Decision-makers had to be supported to take appropriate actions. We developed a 3D visualization system associating geographical information (e.g., a kind of Google Earth representation of the Fukushima environment evolving in real time) and artificial reality objects floating on top of it (e.g., virtual representations of plants, showing available parameters, ongoing rescue reports, radioactivity propagation and so on) (Stephane 2013a,b).

In these three different domains (i.e., space exploration, weather situation awareness and crisis management), we are experiencing the same kind of problems related to the degree of flexibility, innovation, complexity, maturity, stability and sustainability of the technology being designed and developed. Even if software is becoming easier to develop, tangibility of resulting systems must be better understood and mastered. This is impossible without participatory design, agile development and formative evaluations.

## 8. Discussion

It is very clear that 20th century technology-centered engineering followed by human factors investigations and corrective ergonomics is no longer a satisfactory solution for the design and development of current products. 21st century HCD puts people first from the very beginning of the design process, along the entire life cycle of the product.

HCD advocates the search for emerging properties instead of sticking to the currently established systems engineering practice using block diagrams, where people are represented by 'black boxes' linked to other system boxes. These boxes-and-arrows diagrams are useful but dangerous because they assume that people are linear rational systems. More specifically, complexity analysis requires systems thinking looking for emerging behavior and properties (Checkland 1981; Jackson 2003; Daniel-Allegro & Smith 2016). In addition, HCD deals with organizational issues, and therefore needs to be supported by appropriate models (e.g., the Orchestra model).

HCD of complex systems leads to the concept of TIS. A TIS includes both software and hardware. For this reason, the TIS concept can be related to the Internet of Things (IoT), which was coined by Kevin Ashton, an English entrepreneur, to capture the concept of integration between computer-based systems and the physical world (Gardian 1999). Smart phones, smart grid and smart houses are things in the IoT. In the IoT, things have sensors, effectors

and are capable of information processing. TISs and IoT concepts are also very close to Cyber Physical Systems (CPSs), which are systems of embedded systems (Wolf 2014). CPSs are engineered systems that are built from, and depend upon, the seamless integration of computational algorithms and physical components (Lee 2008). The concept of CPS is not new. Most avionics systems in aircraft can be qualified as CPSs. For example, we can find the same kinds of systems in chemical and energy process industries, medicine, automotive, road infrastructure, robotics, and entertainment. Both the IoT and CPSs provide concrete approaches and tools for the development of TISs. The former starts from computer science and information technology premises. The latter starts from physical engineering and automatic control premises. It is interesting to follow the evolution of TISs from both perspectives, and they cross-fertilize each other.

## 9. Conclusion

HCD of complex systems developed over the last decades using automatic control, artificial intelligence, human–computer interaction and systems engineering, as well as human factors and ergonomics, human–computer interaction and HSI. HCD is a discipline that educates and trains *HSI architects* who design recommendations and requirements that engineers develop and manufacture. Human-centered designers, as HSI architects, require creativity skills and the ability to master tangibility.

HCD falls into the domain of open research, where experience is gathered from initiatives and proactive explorations of complex systems design approaches. We need to learn by doing. HCD should be practiced in order to generate such experience. The *task–activity distinction* is at the heart of HCD, a discipline, that is currently young and still under development (i.e., in addition to HCD methods that are predefined tasks, we need to explore HCD productions that are effective activities using the HCD approach).

Content presented in this article is still a work in progress. It is based on more than thirty-five years of work in human–machine systems, exploring human factors, ergonomic solutions, interaction design and more recently HSI. HCD could not have become effective if *modeling and simulation* did not evolve as it did, providing more acceptable realism at design time to use HITLS. In addition, *complexity analysis* is a key process in HCD, which provides a framework and methods for exploration, observation, awareness and rationalization of emerging behaviors and properties of new systems being operated. Finally, *organization design and management* is crucial because HCD is impossible if the organization is not prepared to do it.

## References

- Bainbridge, L.** 1983 Ironies of automation. *Automatica* **19**, 775–780.
- Baxter, G. & Sommerville, I.** 2010 Socio-technical systems: from design methods to systems engineering. *Interacting with Computers* **23** (1), 4–17.
- Bergson, H.** (written in 1889–document published in 2001). *Time and Free Will: An Essay on the Immediate Data of Consciousness* 1910. (Essai sur les données immédiates de la conscience 1889) Dover Publications: ISBN 0-486-41767-0.

- Bødker, S., Ehn, P., Kammersgaard, J., Kyng, M. & Sundblad, Y.** 1987 A Utopian experience. In *Computers and Democracy – A Scandinavian Challenge* (ed. **G. Bjerknes, P. Ehn & M. Kyng**), pp. 251–278. Avebury.
- Boulnois, S. & Boy, G. A.** 2016 Onboard weather situation awareness system: a human–systems integration approach. In *Proceedings of the International Conference on Human–Computer Interaction in Aerospace, Paris, France, September 14–16*. ACM.
- Boy, G. A.** 1997 Active design documents. In *DIS'97 Proceedings – Designing Interactive Systems ACM Conference, Amsterdam, NL – 19–21 August*, pp. 31–36. ACM Press. Also, in the ACM Digital Library.
- Boy, G. A.** 1998 *Cognitive Function Analysis*. Praeger/Ablex. ISBN 9781567503777.
- Boy, G. A.** 2003 *L'Ingénierie Cognitive: Interaction Homme-Machine et Cognition*. (The French Handbook of Cognitive Engineering), Hermes Sciences, Lavoisier.
- Boy, G. A.** 2005 Knowledge management for product maturity. In *Proceedings of the International Conference on Knowledge Capture (K-Cap'05), Banff, Canada, October*. Also, in the ACM Digital Library.
- Boy, G. A.** (Ed.) 2011 *Handbook of Human–Machine Interaction: A Human-Centered Design Approach*. Ashgate.
- Boy, G. A.** 2013 *Orchestrating Human-Centered Design*. Springer. ISBN 978-1-4471-4338-3.
- Boy, G. A.** 2016 *Tangible Interactive Systems: Grasping the Real World with Computers*. Springer. ISBN 978-3-319-30270-6.
- Boy, G. A., Jani, G., Manera, A., Memmott, M., Petrovic, B., Rayad, Y., Stephane, A. L. & Suri, N.** 2016 Improving collaborative work and project management in a nuclear power plant design team: a human-centered design approach. *Annals of Nuclear Energy*. Elsevier. ANE4864.
- Boy, G. A., Mazzone, R. & Conroy, M.** 2010 The virtual camera concept: a third person view. In *Third International Conference on Applied Human Factors and Engineering, Miami, Florida, 17–20 July 2010* (ed. **D. Kaber & G. Boy**). Advances in Cognitive Ergonomics. CRC Press, Taylor & Francis.
- Boy, G. A. & Narkevicius, J.** 2013 Unifying human centered design and systems engineering for human systems integration. In *Complex Systems Design and Management* (ed. **M. Aiguier, F. Boulanger, D. Krob & C. Marchal**). Springer, 2014. ISBN-13: 978-3-319-02811-8.
- Boy, G. A. & Pinet, J.** 2008 *L'être Technologique*. (The Technological Being), L'Harmattan.
- Boy, G. A. & Platt, D.** 2013 A situation awareness assistant for human deep space exploration. In *HCI International Conference, Las Vegas, Nevada, USA, July* (ed. **M. Kurosu**), Human–Computer Interaction, Part IV, HCII 2013, LNCS 8007, pp. 629–636. Springer. Also on the ACM Digital Library (<http://dl.acm.org>).
- Boy, G. A. & Tessier, C.** 1985 Cockpit analysis and assessment by the MESSAGE methodology. In *Proceedings of the 2nd IFAC/IFIP/IFORS/IEA Conference on Analysis, Design and Evaluation of Man–Machine Systems, September 10–12, Villa-Ponti, Italy*, pp. 73–79. Pergamon Press.
- Brown, T.** 2008 Design thinking. *Harvard Business Review* 8592, June.
- Carayon, P.** 2006 Human factors of complex sociotechnical systems. *Applied Ergonomics* 37 (4), 525–535.
- Checkland, P.** 1981 *Systems Thinking, Systems Practice*. John Wiley & Sons.
- Corker, K. M. & Smith, B. R.** 1993 An architecture and model for cognitive engineering simulation analysis: application to advanced aviation automation. In *The Proceedings of the AIAA Computing in Aerospace Conference, October, San Diego, CA*.

- Daniel-Allegro, B. & Smith, G. R.** 2016 Architectural parallels between biological and engineered solutions in defense and security. In *26th Annual INCOSE International Symposium, Edinburgh, UK, July 18–21*.
- Dul, J., Bruder, R., Buckle, P., Carayon, P., Falzon, P., Marras, W. S., Wilson, J. R. & van der Doelen, B.** 2012 A strategy for human factors/ergonomics: developing the discipline and profession. *Ergonomics* 55 (4), 377–395.
- Edwards, E. C. & Kasik, D. J.** 1974 User experience with the CYBER graphics terminal. In *Proceedings of VIM-21, October*, pp. 284–286.
- Fitts, P. M.** (Ed.) 1951 *Human Engineering for an Effective Air Navigation and Traffic Control System*. National Research Council.
- Fitts, P. M. & Jones, R. E.** 1947 Analysis of factors contributing to 460 ‘pilot error’ experiences in operating aircraft controls. Memorandum Report TSEAA-694-12, Aero Medical Laboratory, Air Material Command, Wright-Patterson Air Force Base, Dayton, Ohio, July.
- Gardian** 1999 The Internet of things is revolutionizing our lives, but standards are a must. <http://www.theguardian.com/media-network/2015/mar/31/the-internet-of-things-is-revolutionising-our-lives-but-standards-are-a-must>. Retrieved on July 5, 2015.
- Grudin, J.** 1994 Computer-supported cooperative work: history and focus. *Computer* 27 (5), 19–26.
- Hofweber, T.** 2011 Logic and Ontology. Stanford Encyclopedia of Philosophy. <http://plato.stanford.edu/entries/logic-ontology/#DifConOnt> (retrieved on September 13, 2015).
- Hutchins, E.** 1995 How a cockpit remembers its speeds. *Cognitive Science* 19, 265–288.
- Jackson, M. C.** 2003 *Systems Thinking: Creative Holism for Managers*. John Wiley & Sons.
- Kaptelinin, V.** 1995 Designing learning activity: a cultural-historical perspective in CSCLE. In *Proceedings of the Computer Supported Cooperative Learning (CSCLE’95)*. Indiana University.
- Kaptelinin, V. & Nardi, B.** 2006 *Acting with Technology: Activity Theory and Interaction Design*. MIT Press. ISBN 0-262-51331-5.
- Krzrnaric, R.** 2014 *Empathy: Why It Matters, and How to Get It*. Perigee Books. ISBN-13: 978-0399171390.
- Laurain, T., Boy, G. A. & Stephane, A. L.** 2015 Design of an on-board 3D weather situation awareness system. In *Proceedings 19th Triennial Congress of the IEA, Melbourne, Australia*.
- Lee, E. A.** 2008 Cyber Physical Systems: Design Challenges. <http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-8.html>. Retrieved on May 10, 2015.
- Leifer, L.** 2016 *Dancing with Ambiguity: Embracing the Tension between Divergent and Convergent thinking in Systems Engineering*. Keynote at INCOSE International Symposium, Edinburgh, Scotland (retrieved on 09/05/16): <http://www.incose.org/docs/default-source/events-documents/keynote-monday.pdf?sfvrsn=2>.
- Leont’ev** 1981 *Problems of the Development of the Mind*. Progress.
- Leplat, J.** 1976 Analyse du travail et g n se des conduites. *International Review of Applied Psychology* 25 (1), 3–14.
- Leplat, J. & Hoc, J. M.** 1983 Tache et activit  dans l’analyse psychologique des situations. *Cahiers de Psychologie Cognitive* 3 (1), 49–63.
- Newell, A. & Simon, H.** 1972 *Human Problem Solving*. Prentice-Hall.
- Nietzsche, F.** 1968 *The Will to Power* (ed. W. Kaufmann & R. J. Hollingdale). Vintage. ISBN-13: 978-0394704371.
- Norman, D.** 1988 *The Design of Everyday Things*. Basic Books. ISBN 978-0-465-06710-7.

- Norman, D. A.** 1986 Cognitive engineering. In *User Centered System Design* (ed. **D. A. Norman & S. W. Draper**). Lawrence Erlbaum Associates.
- Norman, D. & Stappers, P. J.** 2016 DesignX: design and complex sociotechnical systems. *She Ji: The Journal of Design, Economics, and Innovation* **1** (2), 83–106. doi:[10.1016/j.sheji.2016.01.002](https://doi.org/10.1016/j.sheji.2016.01.002).
- Papalambros, P. Y. & Wilde, D. J.** 2000 *Principles of Optimal Design: Modeling and Computation*. Cambridge University Press.
- Platt, D.** 2013 The virtual camera: participatory design of a cooperative exploration mediation tool. PhD Dissertation in Human-Centered Design, advised by G. A. Boy, Florida Institute of Technology.
- Platt, D. W. & Boy, G. A.** 2012 The development of a virtual camera system for astronaut-rover planetary exploration. In *Proceedings of the 2012 IEA World Congress on Ergonomics, Recife, Brazil*, pp. 4532–4536. IOS Press. doi:[10.3233/WOR-2012-0032-4532](https://doi.org/10.3233/WOR-2012-0032-4532). Also in *Work* **41** Edited by Marcelo Soares (2012).
- Platt, D. & Boy, G. A.** 2014 Participatory design of the virtual camera for deep space exploration. In *Proceedings of HCI-Aero 2014, Santa Clara, California*. ACM Digital Library.
- Platt, D., Millot, M. & Boy, G. A.** 2013 Design and evaluation of an exploration assistant for human deep space risk mitigation. In *Proceedings of the 12th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems, Las Vegas, Nevada, USA*.
- Plattner, H., Meinel, C. & Leifer, L.** 2016 *Design Thinking – Making Design Thinking Fundamental*. Springer. ISBN 978-3-319-19640-4.
- Poltrock, S. E. & Grudin, J.** 2003 Collaboration Technology in Teams, Organizations, and Communities. *CHI 2003 Tutorial*. <http://research.microsoft.com/~jgrudin>.
- Rasmussen, J.** 1983 Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human-performance models. *IEEE Transactions Systems, Man, and Cybernetics SMC* **13** (3), 257–266.
- Rochlis Zumbado, J.** 2015 *Human-Systems Integration (HSI) Practitioner's Guide*. NASA/SP-2015-3709. Johnson space Center, Houston, TX.
- Sarter, N., Woods, D. D. & Billings, C. E.** 1997 Automation surprises. In *Handbook of Human Factors & Ergonomics*, 2nd edn (ed. **G. Salvendy**). Wiley.
- SCRUM** 2015 <https://www.scrum.org>. (Retrieved on January 26, 2015).
- Stephane, A. L.** 2013a Analysis of the Fukushima disaster: reinforcement for using STAMP as a vector of safety governance. In *Proceedings of the Second STAMP Conference, March 26–28, MIT*.
- Stephane, A. L.** 2013b The art of crisis management and case study: 3D ‘Serious Game’ for Fukushima Daiichi. *PUF Summer School*. Valenciennes, France.
- Suchman, L.** 1987 *Plans and Situated Actions: The Problem of Human-Machine Communications*. Cambridge University Press.
- Sutherland, J.** 2014 *Scrum: The Art of Doing Twice the Work in Half the Time*. Crown Business. ISBN 978-0385346450.
- Tan, W. & Boy, G. A.** 2016 Iterative designs of Onboard Context-Sensitive Information System (OCSIS). In *Proceedings of the International Conference on Human-Computer Interaction in Aerospace, Paris, France, September 14–16*. Also in the ACM Digital Library.
- Varela, F. J., Thompson, E. & Rosch, E.** 1999 *The Embodied Mind*. Seventh Printing. MIT Press.

- Vicente, K. J.** 2002 Ecological interface design: progress and challenges. *Human Factors* 44 (1), 62–78.
- Weinberg, U.** 2012 Querdenken im Team – Mit Design Thinking wird Innovation zur Routine. In *Smarte Innovation – Ergebnisse und neue Ansätze im Maschinen- und Anlagenbau* (ed. S. Pfeiffer, P. Schütt & D. Wühr), pp. 247–252. Springer Fachmedien Wiesbaden GmbH.
- Wickens, C. D., Lee, J. D., Liu, Y. & Gordon-Becker, S.** 2003 *Introduction to Human Factors Engineering*. Pearson. ISBN 978-0131837362.
- Wiener, E. L.** 1989 Human factors of advanced technology ('glass cockpit') transport aircraft. (NASA Contractor Report No. 177528). Moffett Field, CA: NASA-Ames Research Center.
- Winograd, T. & Flores, F.** 1987 *Understanding Computers and Cognition: A New Foundation for Design*. Addison-Wesley. ISBN-13: 978-0201112979.
- Wolf, M.** 2014 *High-Performance Embedded Computing, Second Edition: Applications in Cyber-Physical Systems and Mobile Computing*, 2nd edn Morgan Kaufmann. ISBN-13: 978-0124105119.