Introduction to Human-Machine Interaction: A Human-Centered Design Approach

Guy A. Boy

Rationale

Nobody questions the use of the clock today: the main function of a clock is to provide the time to its user. A modern watch uses several resources that include a battery, internal mechanisms and the ability of its user to (re-)adjust time when necessary or change the battery when it is no longer working. You interact with your watch as you would with someone who will tell you “hey, it is time to go to the next meeting!” This automaton can be programmed and consequently act as an agent that supports many time-related facets of your life. More generally, automation brought up and consolidated the concept of human-machine interaction (HMI).

HMI, as a field of investigation, is quite recent even if people have used machines for a long time. HMI attempts to rationalize relevant attributes and categories that emerge from the use of (computerized) machines. Four main principles, i.e., safety, performance, comfort and esthetics, drive this rationalization along four human factors lines of investigation: physical (i.e., physiological and bio-mechanical), cognitive, social or emotional.

Physically, the sailor interacts with his/her boat by pulling sail ropes for example. Cognitively, I interact with my computer writing the introduction of this book. Of course I type on a keyboard and this is physical, but the main task is cognitive in the sense that I need to control the syntax and the semantics of my writing, as well as spelling feedback provided by my text processing application. Software makes it more cognitive. You may say that the sailor needs to know when and how to pull the ropes, and this is a cognitive activity. Indeed, learning is required to optimize workload among other human factors. Socially, it happens that my colleague and I wrote this text for a community of people. Any production, which is targeted to a wider audience than its producer could anticipate, becomes a social production that will need to be socially accepted. This is true for an engineering production, but also for a legal act or an artistic production. Emotionally, the artist uses his/her pen or computer to express his/her emotions. But, emotions may come from situations also where adrenalin is required to handle risky decisions and actions. More generally, esthetics involves empathy in the human-machine relation (Boy & Morel, 2004).

For the last three decades, cognition was central to the study of human-machine interaction. This is because automation and software mediates most tasks. Hollnagel and Woods (2005), talking about the growing complexity of interaction with increasingly computerized systems, introduced this concept of changing balance between doing and thinking. But, what do we mean by “doing” today when we permanently use computers for most of our everyday tasks. Doing is interacting… with software! HMI has become human-computer interaction (HCI).
However, in this book, HMI is more than HCI even if it includes it. Driving a car, flying an airplane or controlling a chemical plant is obviously not the same as text processing. In this sense, HMI is not the same as HCI (Hewett et al., 1992; Card et al., 1983; Myers, 1998; Sears & Jacko, 2007).

HMI has become a mandatory field of research and engineering in the design and development of nowadays systems. Why? As already said, this is because what is commonly called a user interface is currently made of software… and this user interface has become deeper and deeper! We now interact with machines through software. The main issue is to develop systems that enable appropriate task execution through them. We often think that we simplify tasks by piling layers of software, but it happens that resulting interaction is sometimes more complicated and complex. Indeed, even if HCI strongly contributes to decrease interaction complexity, the distance between people and the physical world increases so much that situation awareness becomes a crucial issue, i.e., we must not lose the sense of reality.

Therefore, what should we do? Should we design simpler systems that people would use without extensive learning and performance support? To what extent should we accept the fact that new systems require adaptation from their users? Where should we draw the line? An important distinction is between evolution and revolution. For example, cars are getting more computerized, e.g., in addition to the radio, new systems were introduced such as the global positioning system (GPS) that supports navigation, an autopilot in the form of a speed control system and a line keeping system, an onboard computer that support energy consumption, a collision avoidance system, an hand-free kit that enables the driver to communicate with people located outside of the vehicle, and so on.

Even if the initial goal was to improve safety, performance, comfort and esthetics, the result today is that there are far too many onboard systems that increase driver’s workload and induce new types of incidents and accidents. On one side, software technology attempts to help people, and on the other side, it induces new types of life-critical problems that any designer or engineer should take into account in order to develop appropriate solutions. A simple alarm provided by the central software of a modern car may end up in a very complicated situation because neither the driver nor a regular mechanic will be able to understand what is really going on; a specialized test machine is required together with the appropriate specialized person who knows how to use it.

This handbook proposes approaches, methods and tools to handle HMI at design time. For that matter, it proposes a human-centered design (HCD) approach. Of course, it must be considered as a starter toward a deeper search into the growing bulk of HMI and HCD literature and field. It is based on contemporary knowledge and know-how on human-machine interaction and human-centered design. It is targeted at a diverse audience including academia and industry professionals. In particular, it should serve as a useful resource for scholars and students of engineering, design and human factors, whether practitioners or scientists, as well as members of the general public with an interest in cognitive engineering (Norman, 1982, 1986), cognitive system engineering (Hollnagel & Woods, 1983, 2005) and human-centered design (Boy, 1998). Above all, the volume is designed as a research guide that will both inform readers on the basics of human-machine interaction, and provide a look ahead at the means through which cognitive engineers and human factors specialists will attempt to keep developing the field of HMI.
Human-Centered Automation

Current machines heavily rely on the cognitive skills of their users, who acquire and process data, make decisions and act in real-time. In particular, we will focus on the use of automated machines and various kinds of innovative artifacts developed to overcome limitations of humans facing the growing complexity of their overall environment. Automation will never stop\(^1\). This is currently the case of the evolution of air traffic management (ATM) where controllers have to face the difficult challenge of managing a growth of 4.5% per year average for the last twenty years; this growth is very likely to remain the same during the next decades. Machines are becoming more complex even if the goal of the designers is to facilitate their use during normal operations, problems happens in routine as well as abnormal contexts. This is why human reliability needs to be taken carefully from tow points of view: (1) humans have limitations; (2) humans are unique problem-solvers in unanticipated situations.

Cognitive engineering should better benefit from operational experience by proposing appropriate models of cognition that would rationalize this experience. Rasmussen’s model has been extensively used over the last two decades to explain the behavior of a human operator controlling a complex dynamic system (Rasmussen, 1986). This model is organized into three levels of behavior: skill, rule and knowledge (Figure 1).

Historically, automation of complex dynamic systems, aircraft in particular, have led to the transfer of human operators’ skills (e.g., performing a tracking task) to the machine. Autopilots have been in charge of simple tracking tasks since the 1930s. This kind of automation was made possible using concepts and tools from electrical engineering, mechanical engineering and control theories, such as mechanical regulators, proportional-integral-derivative controllers (PID), Laplace functions and stochastic filtering. Autopilots were deeply refined and rationalized during the sixties and the seventies. Human skill models were based on quasi-linear models’ functions and optimal control models. Human engineering specialists have developed quantitative models to describe and predict human control performance and workload at Rasmussen's skill level. They have been successfully applied to study a wide range of problems in the aviation domain such as handling qualities, display and control system design and analysis, and simulator design and use.

The second automation revolution took place when the rule-based level was transferred to the machine. In aviation, a second layer was put on top of the autopilot to take care of navigation. The flight management system (FMS) was designed and implemented during the eighties to provide set points for the autopilot. A database is now available onboard with a large variety

\(^1\) Bernard Ziegler, a former Vice-President of Airbus Industrie, made the following observations and requirements from his experience as a test pilot and distinguished engineer: “the machine that we will be handling will become increasingly automated; we must therefore learn to work as a team with automation; a robot is not a leader, in the strategic sense of the term, but a remarkable operator; humans will never be perfect operators, even if they indisputably have the capabilities to be leaders; strategy is in the pilot’s domain, but not necessarily tactics; the pilot must understand why the automaton does something, and the necessary details of how; it must be possible for the pilot to immediately replace the automaton, but only if he has the capability and can do better; whenever humans take control, the robot must be eliminated; the pilot must be able to trust automation; acknowledge that it is not human nature to fly; it follows that a thinking process is required to situate oneself, and in the end, as humiliating as it may be, the only way to insure safety is to use protective barriers.” (Ziegler, 1996).
of routes that cover most of the flights in a specific geographical sector. Pilots need to program the FMS by recalling routes from the database and eventually customize them for a specific flight. Once they have programmed the FMS, the aircraft is “capable of flying by itself” under certain conditions, i.e., the FMS is in charge of the navigation task in pre-programmed situations.

![Figure 1. Rasmussen’s model, automation evolution and contributing discipline emergence.](image)

Today, human operators mostly work at Rasmussen’s knowledge-based level where interpretation has become an important work process. Basic operations are delegated to the machine, and humans progressively become managers of (networked) cognitive systems. Humans need to identify a situation when there is no pattern matching (situation recognition) at the rule-based level, to decide according to specified (or sometimes unspecified) goals, and to plan a series of tasks. These are typical strategic activities. Some people are good at strategic activities, others prefer to execute what they are told to do. In any case, the control of cognitive systems requires strategic training. For example, using the Web has totally transferred the shopping task to Rasmussen's knowledge-based level, i.e., the selection of food items is made using virtual objects, and delivery is planned with respect to the customer’s schedule and the nature of the food.

Technology has always contributed to shape the way people interact with the world. Conversely, interacting with the world has direct impact on how technology evolves. Rationalization of experience feedback influences the development of theories that make new artifacts emerge. In a technology-driven society, this goes the other way around, i.e., the use of artifacts induces new practices, and new jobs emerge, as film technology induced the art of film making for example. The twentieth century was rich in technology innovation and development. The speed of evolution of technology and resulting practices is very sensitive to economical impacts. In some cases, when economical benefits were not obvious a priori, but the evolution of human kind was at stake, technological advances were decided at the political level, such as designing and developing a technology that enables a man to walk on the moon. Today following these grandiose projects, we realize that human-centered automation, and
more generally human-centered design and engineering, is not enough effectively taken into account at the political level yet. A new paradigm needs to be found to better understand the optimal allocation between human and machine cognition together with the evolution of organizations.

The term “human-centered automation” (HCA) was coined by Billings (1991) in the aviation domain, and, among a large variety of research efforts, further developed (Boy et al., 1995). Human-centeredness requires that we focus on some distinctions. When it was conceived, HCA differed from human-centered design (HCD) in the sense that automation is something added to an existing system. Since software technology is dominant in the systems that we develop today, HCA becomes necessarily HCD. But, I think that there is an even more important distinction between HCD and human-centered engineering (HCE). HCD is the mandatory upstream process that enables a design team to incorporate human requirements into the design of a system. Usually, HCD is scenario-based and prototype-based. It consists in gathering human factors issues from an appropriate community of users or, more generally, actors who are anticipated to act on the system being designed. These actors may be direct end-users but also maintainers who will have to repair the system in case of failure for example. In this case, it is not only design for usability, but also design for maintainability. At this stage, we need to investigate possible scenarios that make actors requirements as explicit as possible. In the same way, and as architects do for the design of buildings, mock-ups need to be developed in order to incrementally validate actors requirements (this is formative evaluation). HCE follows HCD. Human factors engineers are required to check the various steps of the production of the system. If HCD is creativity-driven, HCE is a systematic process that is based on a body of rules that need to be applied. In aeronautics, HCE is now a common practice and is formalized by official national and international regulatory institutions, e.g., ISO\textsuperscript{2}, ICAO\textsuperscript{3}, IATA\textsuperscript{4} and EASA\textsuperscript{5}. Examples of such rules are provided in EASA CS.25-1302 (2004) and ISO 13407 (1999). In this book, we will mainly focus on HCD, even if some of the chapters treat parts of currently-practiced HCE, and insist on the fact that end-user expertise and experience should be used during the whole life cycle of any artifact.

**The AUTOS Pyramid**

The AUTOS pyramid is a framework that helps rationalize human-centered design and engineering. It was first introduced in the HCD domain as the AUTO tetrahedron (Boy, 1998) to help relate four entities: Artifact (i.e. system), User, Task and Organizational environment. Artifacts may be aircraft or consumer electronics systems, devices and parts for example. 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

\textsuperscript{2} International Standard Organization.

\textsuperscript{3} International Civil Aviation Organization.

\textsuperscript{4} International Air Transport Association.

\textsuperscript{5} European Aviation Safety Agency.
learn and use. The AUT triangle (Figure 2) enables the explanation of three edges: task and activity analysis (U-T); information requirements and technological limitations (T-A); ergonomics and training (procedures) (T-U).

![AUT triangle](image)

*Figure 2. The AUT triangle.*

The organizational environment includes all team players who/that will be called “agents”, whether humans or machines, interacting with the user who performs the task using the artifact (Figure 3). It introduces three additional edges: social issues (U-O); role and job analyses (T-O); emergence and evolution (A-O).

![AUTO tetrahedron](image)

*Figure 3. The AUTO tetrahedron.*

The AUTOS framework (Figure 4) is an extension of the AUTO tetrahedron that introduces a new dimension, the “Situation”, which was implicitly included in the “Organizational environment”. The three new edges are: usability/usefulness (A-S); situation awareness (U-S); situated actions (T-S); cooperation/coordination (O-S).

HMI could be presented by describing human factors, machine factors and interaction factors. Using AUTOS, human factors are user factors, machine factors are artifact factors, and interaction factors combine task factors, organizational factors and situational factors. Of course, there are many other ways to present this discipline, we choose the five AUTOS dimensions because they have been proved to be very useful to drive human-centered design and categorize HMI complexity into relevant and appropriate issues. Therefore, I use them to structure this introduction of HMI. These aspects include design methods, techniques and tools.
Machine factors (the A of AUTOS)

Since this book is devoted to support designers and engineers in the design and development of human-machine systems, technological aspects are important concrete bricks that will enable them to perform their jobs. In this handbook, machine factors will not be developed from an engineering viewpoint, but a usage viewpoint.

Hardware factors

Today, people at work typically face computer screens of various forms and use a variety of control devices. We usually refer to the user interface, i.e., a system in between a user (or human operator) and a process to be controlled or managed. The terms “human operator” and “user” can be equally used. The former come from process control and the human-system engineering community. The latter comes from the human-computer interaction (HCI) community. In these two communities, automation took various forms and contents. In process control, automation was driven by control theories where feedback is the dominant concept. In HCI, (office) automation was driven by the desktop metaphor for a long time to the point out that usability often refers to the ability to use a graphical interface that includes menus, buttons, windows and so on. It is interesting to note that the process control discipline took care of real-time continuous processes such as nuclear, aerospace or medical systems where time and dynamics are crucial issues together with safety-critical issues. Conversely, HCI developed the interaction comfort side. HCI specialists got interested into learnability, efficiency, easy access to data, direct manipulation of metaphoric software objects and pleasurable user experience. Human operators are typically experts because industrial processes that they control are complex and safety-critical. Users, in the HCI sense, can be anybody.

For all these reasons, hardware factors are different if we talk about process control or “classical” HCI. The very nature of processes to be controlled needs to be analyzed and understood well enough to determine what kind of hardware would be suitable for the safest, most efficient and comfortable human-machine interaction. However, HCI strongly influenced our lives during the last decade to the point that usability has become a standard even in process control interface design and development. People are now very familiar with menus and windows. This is fine, but this also assumes that these interaction styles cover the various constraints of the process being controlled.
I recently heard the term “interactive cockpit”. What does that mean? I always thought that an aircraft cockpit was the ultimate interactive interface with a dynamic safety-critical process, and therefore interactive. But “interactive” means something else today. It does not mean continuous interaction through a steering wheel or a handle physically connected to rudders, it means interacting with a piece of software… typically through a computer screen with a pointing device and a keyboard! This (r)evolution started with the glass cockpits in the mid-eighties; we were talking about “fly-by-wire”. Today, the car industry is talking about “drive-by-wire”, but the meaning has also changed following this technological evolution where software is the most important component.

There are hardware factors that incrementally emerge from the design of these new machines. Among them, the following are important: force feedback, loudspeakers, screens, signals, buttons, keyboard, joystick, mouse, trackball, microphone, 3D mouse, data glove, data suit (or interactive seat), metaphor for interaction, visual rendering, 3D sound rendering, 3D geometrical model and so on.

From the time of the first flying machines at the end of the nineteen-century to the Concorde, the number of instruments in an aircraft cockpit grew up to 600. At the beginning of the eighties, the introduction of cathode ray tubes (CRT) and digital technology in cockpits contributed to drastically decrease this number. Today, the number of displays in the A380 is considerably reduced. This does not mean that the number of functions and lines of software code is reduced. As a matter of fact, software size keeps increasing tremendously.

**Software factors**

Software is very easy to modify; consequently we modify it all the time! Interaction is not only a matter of end product; it is also a matter of development process. End-users are not the only ones to interact with a delivered product; designers and engineers also interact with the product in order to fix it up toward maturity… even after its delivery. One of the reasons is that there are software tests that cannot be done without a real-world exposure. It is very difficult to anticipate what end-users would do in the infinite number of situations where they will be evolving. We will see in this book that scenario-based design is mandatory with respect to various dimensions including understandability (situation awareness), complexity, reliability, maturity and induced organizational constraints (rigidity versus flexibility).

What should we understand when we use a product? How does it work? How should it be used? At what level of depth should we go inside the product to use it appropriately? In the early ages of the car industry, most car drivers were also mechanics because when they had a problem they needed to fix it by themselves; the technology was too new to have specialized people. These drivers were highly skilled engineers both generalists and specialists on cars. Today, things have drastically changed; drivers are no longer knowledgeable and skilled to fix cars; there are specialists that do this job because software is far too complex to understand without appropriate help. Recent evolution transformed the job of mechanics into system engineers who know how to use specialized software that enables to diagnose failures and fix them. They do not have to fully understand what is going on inside the engine, a software program does it for them and explain problems to them; when the overall system is well-designed of course. This would be the ideal case; in practice, most problems come from organizational factors induced by the use of such technology, e.g., appropriate people may not be available at the right time to fix problems when they arise.
Software complexity can be split into internal complexity (or system complexity) and interface complexity. Internal complexity is related to the degree of explanation required to the user to understand what is going on when necessary. Concepts related to system complexity are: flexibility (both system flexibility and flexibility of use); system maturity (before getting mature, a system is an accumulation of functions —the “another function syndrome”— and it becomes mature through a series of articulations and integrations); automation (linked to the level of operational assistance, authority delegation and automation culture); and operational documentation. Technical documentation complexity is very interesting to be tested because it is directly linked to the explanation of artifact complexity. The harder a system is to use, the more related technical documentation or performance support are required in order to provide appropriate assistance at the right time in the right format.

In any case, software should be reliable at any time in order to support safe, efficient and comfortable work. There are many ways to test software reliability (Lyu, 1995; Rook, 1990). In this handbook, what we try to promote is not only system reliability, but also human-machine reliability. We know that there is a co-adaptation of people and machines (via designers and engineers). Human operators may accept some unreliable situations where the machine may fail as long as safety, efficiency and comfort costs are not too high. However, when these costs become high enough for them, the machine is just rejected. Again this poses the problem of product maturity (Boy, 2005); the conventional capacity maturity model for software development (Paulk et al., 1995), systematically used in most industries, does not guarantee product maturity, but process maturity. Product maturity requires continuous investment of end-users in design and development processes. At the very beginning, they must be involved with domain specialists to set up high-level requirements right; this is an important role of participatory design. During the design and development phase, formative evaluations should be performed involving appropriate potential end-users in order to “invent” the most appropriate future use of the product.

Interface complexity is characterized by content management, information density and ergonomics rules. Content management is, in particular, linked to information relevance, alarm management, and display content management. Information density is linked to decluttering, information modality, diversity, and information-limited attractors, i.e., objects on the instrument or display that are poorly informative for the execution of the task but nevertheless attract user’s attention. The “PC screen do-it all syndrome” is a good indicator of information density (elicited improvement-factors were screen size and zooming). Redundancy is always a good rule whether it repeats information for crosschecking, confirmation or comfort, or by explaining the “how”, “where”, and “when” an action can be performed. Ergonomics rules formalize user friendliness, i.e., consistency, customization, human reliability, affordances, feedback, visibility and appropriateness of the cognitive functions involved. Human reliability involves human error tolerance (therefore the need for recovery means) and human error resistance (therefore the existence of risk to resist to). To summarize, A-factors deal with the level of necessary interface simplicity, explanation, redundancy and situation awareness that the artifact is required to offer to users.
Human factors (the U of AUTOS)

Human factors have been heavily studied during the last five decades in the HMI context. After the Second World War, human factors specialists were mainly physicians, medical doctors, who were taking care of both physiological and biomechanical aspects of humans at work, i.e., ergonomics. Work psychology, and later on cognitive psychology, progressively emerged in the seventies to become the leading discipline in human factors in the eighties. The main reason of this emergence is the introduction of computers and software in work places and, more generally, everyday life. All these approaches were essentially based on the Newell and Simon’s information processing model that is typically a single agent model (Newell & Simon, 1972). The development of computerized systems and more specifically networked systems promoted the need for studying social and organizational factors. We have then moved into the field of multi-agent models.

HMI involves automation, i.e., machines that were controlled manually are now managed through a piece of software that mediates user intentions and provides appropriate feedback. Automation introduces constraints, and therefore rigidity. Since end-users do not have the final action, they need to plan more than in the past. As already said, work becomes more cognitive and (artificially) social, i.e., there are new social activities that need to be performed in order for the other relevant actors to do their jobs appropriately. This even becomes more obvious when cognition is distributed among many human and machine agents. Computer-supported cooperative work, for example, introduced new types of work practices that are mandatory to learn and use, otherwise overall performance may rapidly become a disaster.

Human body-related and physiological factors

Work performed by people can be strongly constrained by their physiological and biomechanical possibilities and limitations. Understanding these possibilities and limitations tremendously facilitated the evolution of civil and military aviation. Cockpits were incrementally shaped to human anthropometrical requirements in order to ease manipulation of the various instruments. This of course is always strongly related to technology limitations also.

Anthropometry developed its own language and methods. It is now actively used in design to define workspaces according to human factors such as accommodation, compatibility, operability, and maintainability by the user population. Workspaces are generally designed for 90% to 95% coverage of the user population. Anthropometric databases are constantly maintained to provide appropriate information to designers and engineers. Nevertheless, designers and engineers need to be guided to use these databases in order to make appropriate choices.

Work organization is also a matter of trouble for professionals. Fatigue is a major concern. Therefore, it is important to know about circadian rhythms and the way people adapt to shift work and long work hours for example. Consequences are intimately associated with health

---

6 Professor Grandjean declared founded the International Ergonomics Association (IEA) on April 6, 1959, at a meeting in Oxford, England. Today, IEA is a hyper-organization that has 42 federated societies, 1 affiliated society, 11 sustaining member organizations, 6 individual sustaining members and 2 networks. It also has 22 Technical Committees (http://www.iea.cc). IEA includes all forms of human factors at work.
and safety risks. Fatigue studies provide more knowledge and knowhow on how to proceed with work time schedules, appropriate training, systematic checks, and health indicators following. Of course, this needs to be integrated in regulatory procedures.

**Cognitive factors**

Cognitive factors start with workload assessment. This statement may seem to be restrictive and old fashion, but the reader should think twice about workload before starting any work in human factors. On one side, workload is a concept that is very difficult to define. It is both an output of human performance and a necessary input to optimize performance, i.e., we produce workload to perform better, up to a point where we need to change our work strategy. But on the other side, we need to figure out a model that would quantify a degree of load produced by a human being while working. Of course, this model should be based on real measurements performed on the human being. Many models of workload have been proposed and used (Bainbridge, 1978; Hart, 1982; Boy & Tessier, 1985). Workload also deals with the complexity of the task being performed. In particular, people can do several things at the same time, in parallel; this involves the use of several different peripheral resources simultaneously (Wickens, 1984). Sperandio (1972) studied the way air traffic controllers handle several aircraft at the same time, and showed that the time spent on radio increased with the number of aircraft being controlled: 18% of their time spent in radio communication for one controlled aircraft whereas 87% for nine aircraft controlled in parallel. In other words, task complexity tends to increase human operator efficiency.

Human-machine interaction moves into human-machine cooperation when the machine becomes highly automated. In this case, it is more appropriate to talk about agent-agent cooperation. Hoc and Lemoine studied dynamic task allocation (DTA) of conflict resolution between aircraft in air-traffic control on a large-scale simulator. “It included three cognitive agents: the radar controller (RC), in charge of conflict detection and resolution; the planning controller (PC), in charge of entry-exit coordination and workload regulation; and a conflict resolution computer system. Comparisons were made on the basis of a detailed cognitive analysis of verbal protocols. The more the assistance, the more anticipative the mode of operation in controllers and the easier the human-human cooperation (HHC). These positive effects of the computer support are interpreted in terms of decreased workload and increased shared information space. In addition, the more the controllers felt responsible for task allocation, the more they criticized the machine operation” (Hoc & Lemoine, 1998).

Situation awareness (SA) is another concept that is useful to introduce here, especially as a potential indicator for safety in highly automated human-machine systems. During the last decade, lots of efforts have been carried out to assess SA such as the Situation Awareness Global Assessment Technique (SAGAT) (Endsley, 1987, 1996). Several efforts have been developed to assess SA in the aeronautics domain (Mogford, 1997; Stephane & Boy, 2005); the main problem is the characterization of the influence of action on situation awareness. Indeed, human operator’s actions are always situated, especially in life-critical environments, and SA does not mean the same when actions are deliberate as when they are reactive. In human-machine interaction, this is a very important issue since actions are always both intentional (deliberative) and reactive because they are mainly performed in a close loop.

Obviously, there are many other concepts and processes that need to be taken into account to investigate cognitive factors. I would like to insist on the fact that cognition must be thought
in a multi-agent perspective where human-machine interaction is in fact a dynamic network of interactions among human and machine agents. Consequently, cognition should be taken more broadly than in cognitive psychology and extend it to a social and organizational perspective.

**Social factors**

There are two fields of research that grew independently for the last three decades: crew resource management (CRM) in aviation, and computer-supported cooperative (CSCW) work in HCI. The former was motivated by social micro-world of aircraft cockpits where pilots need to cooperate and coordinate to fly safely and efficiently. CRM started during a workshop on *resource management on the flight deck* sponsored by NASA in 1979 (Cooper, White, & Lauber, 1980). At that time, the motivation was the correlation between air crashes and human errors as failures of interpersonal communications, decision-making, and leadership (Helmreich et al., 1999). CRM training developed within airlines in order to change attitudes and behavior of flight crews. CRM deals with personalities of the various human agents involved in work situations, and is mainly focused on teaching, i.e., each agent learns to better understand his or her personality in order to improve the overall cooperation and coordination of the working group.

Douglas Engelbart is certainly the most influential contributor to the development of the technology that supports collaborative processes today. He invented the mouse and worked on the ARPANET project in 1960s. He was among the first researchers who developed hypertext technology and computer networks to augment intellectual capacities of people. The term “computer-supported cooperative work” (CSCW) was coined in 1984 by Paul Cashman and Irene Grief to describe a mutli-disciplinary approach focused on how people work and how technology could support them. CSCW scientific conferences were first organized in the USA within the ACM-SIGCHI community. Conferences on the topic immediately followed in Europe and Asia. Related work and serious interest already existed in European Nordic countries. During the late 1970s and even more during the 1980s, office automation was born from the emergence of new practices using minicomputers. Minicomputers and microcomputers were integrated in many places such as travel agencies, administrations, banks and so on, to support groups and organizations. People started to use them interactively, as opposed to using them in a batch mode. Single user applications such as text processors and spreadsheets were developed to support basic office tasks. Several researchers started to investigate the way people were using this new technology. Computer science is originally the science of internal functions of computers (how computers work). With the massive use of computers and their incremental integration in our lives, computer science has also become the science of external functions of computers (how to use computers and what they are for). We, computer and cognitive scientists, needed to investigate and better understand more how people appropriate computers individually and collectively to support collaborative work. Multi-disciplinary research developed involving psychologists, sociologists, education and organization specialists, managers and engineers.

In parallel with these two fields of research, two others developed: human reliability (Reason, 1990) and distributed cognition (Hutchins, 1995). The former led to a very interesting

---

7 Association for Computing Machinery – Special Interest Group on Computer Human Interaction.
distinction between two approaches of human reliability whether the focus is on the person or the system. Each approach induces a quite different philosophy of error management from the other. Reason developed what he called the Swiss cheese model (Reason, 1997). He stated that we cannot change human limitations and capabilities, but the conditions in which humans perform their tasks can be changed. Therefore, these conditions, which can be viewed as technological and organizational constraints, should be clearly identified in order to create defensive barriers against the progression of an unsafe act.

Distributed cognition was first developed to take into account the sharing of meaningful concepts among various agents. Extending the phenomenological school of thought, agents are considered as subjects and not objects. They have different subjectivity, and therefore they need to adapt among each other in order to develop a reasonable level of empathy, consensus and common sense sharing; this is what intersubjectivity is about: “The sharing of subjective states by two or more individuals” (Scheff 2006). This line of research cannot avoid taking into account intercultural specificities and differences. It is not surprising that most leaders of such a field come from anthropology and ethnology. Obviously, the best way to better understand interactions between cognitive agents is to be integrated in the community of these agents. In the framework of human-machine systems, we extend the concept of distributed cognition to humans and machines. The extension of the intersubjectivity concept to humans and machines requires that we take into account end-users and designers in a participatory way. To summarize, U-factors mainly deal with *user’s knowledge, skills and expertise* on the new artifact and its integration.

**Interaction factors (the TOS of AUTOS)**

**Task factors**

Human-machine interaction is always motivated by the execution of a task. Therefore, the way the task is organized and supported by the machine (prescribed task), and executed by the human user (effective task) is crucial. Obviously the effective task, that is often called “activity”, is different from the prescribed task.

Activity analysis could be defined as the “identification and description of activities in an organization, and evaluation of their impact on its operations. Activity analysis determines (1) what activities are executed, (2) how many people perform the activities, (3) how much time they spend on them, (4) how much and which resources are consumed, (5) what operational data best reflects the performance of activities, and (6) of what value the activities are to the organization. This analysis is accomplished through direct observation, interviews, questionnaires, and review of the work records. See also job analysis, performance analysis and task analysis.” (Business Dictionary, 2009)

Task complexity involves procedure adequacy, appropriate multi-agent cooperation (e.g., air-ground coupling in the aerospace domain) and rapid prototyping (i.e., task complexity cannot be properly understood if the resulting activity of agents involved in it is not observable). Task complexity is linked to the number of sub-tasks, task difficulty, induced risk, consistency (lexical, syntactic, semantic and pragmatic) and the temporal dimension (perception-action frequency and time pressure in particular). Task complexity is due to operations maturity, delegation and mode management. Mode management is related to role
analysis. To summarize, T-factors mainly deal with task difficulty according to a spectrum from best practice to well-identified categories of tasks.

Organizational factors

Interaction is also influenced by the organizational environment that is itself organized around human(s) and machine(s) in the overall human-machine system (HMS). More explicitly, an HMS could be someone facing his/her laptop writing a paper; it could also be someone driving a car with passengers; it could be an air traffic management system that includes pilots, controllers and various kinds of aviation systems. People are now able to interact with computerized systems or with other people via computerized systems. We recently put to the front authority as a major concept in human-centered automation. When a system or other parties do the job, or part of the job, for someone, there is delegation. What is delegated? Is it the task? It is the authority in the execution of this task? By authority, we mean accountability (responsibility) and control.

Organization complexity is linked to social cognition, agent-network complexity, and more generally multi-agent management issues. There are four principles for multi-agent management: agent activity (i.e., what the other agent is doing now and for how long); agent activity history (i.e., what the other agent has done); agent activity rationale (i.e., why the other agent is doing what it does); and agent activity intention (i.e., what the other agent is going to do next and when). Multi-agent management needs to be understood through a role (and job) analysis. To summarize, O-factors mainly deal with the required level of coupling between the various purposeful agents to handle the new artifact.

Situational factors

Interaction depends on the situation where it takes place. Situations could be normal or abnormal. They could even be emergencies. This is why we will emphasize the scenario-based approach to design and engineering. Resulting methods are based on descriptions of people using technology in order to better understand how this technology is, or could be, used to redefine their activities. Scenarios can be created very early during the design process and incrementally modified to support product construction and refinement.

Scenarios are good to identify functions at design time and operations time. They tend to rationalize the way the various agents interact among each other. They enable the definition of organizational configurations and time-wise chronologies.

Situation complexity is often caused by interruptions and more generally disturbances. It involves safety and high workload situations. It is commonly analyzed by decomposing contexts into sub-contexts. Within each sub-context, the situation is characterized by uncertainty, unpredictability and various kinds of abnormalities. To summarize, situational factors deal with the predictability and appropriate completeness (scenario representativeness) of the various situations in which the new artifact will be used.

Overview of the handbook

Of course, hard decisions needed to be made on the main topics that are developed in this handbook. In addition to this introduction and the conclusion, the book is organized into three
parts (analysis; design and engineering; and evaluation) and twenty chapters. These chapters include transversal perspectives on human-machine interaction, methods and tools for human-centered design and engineering, and continuity and change in human-machine systems.

A handbook on human-machine interaction cannot avoid human operator modeling. Thierry Bellet presents an account on analysis, modeling and simulation of human operator’s mental activities. Even if he limits his illustrations to car driving, the various descriptions and methods are applicable to other safety-critical systems. Most readers know what car driving is in practice, therefore examples will be better understood.

Following up on human factors, situation awareness became a major topic over the last two decades mainly because the sophistication of technology tends to increase the distance between human operators and the actual work. Anke Popken and Josef Krems present the relation between automation and situation awareness, and illustrative examples in the automotive domain.

There are several aspects of human factors such as psychophysiology and performance that are certainly very important to take into account at design time. Anil Raj, Margery Doyle and Joshua Cameron present an approach and results that can be used in human-centered design. They focus on the relationships between workload, situation awareness and decision effectiveness.

For the last three decades, human reliability was a hot topic, mainly in aviation, and more generally in life-critical systems. Christopher Johnson presents a very comprehensive approach to human error in the context of human-machine interaction, and more specifically in human-centered design.

Complex systems cannot be operated without operational support. Barbara Burian and Lynne Martin present a very experienced account on operating documents that change in real-time. Software enables the design of interactive documents, electronic flight bags, integrated navigational maps, and electronic checklists for example. New technology enables the integration of information from different sources and ease the manipulation of resulting data in real-time.

Human-machine interaction was thought as a human operator interacting with a machine. Today, the human-machine social environment changes toward multi-agent interactions. Guy Boy and Gudela Grote describe this evolution and the mandatory concepts that emerge form this evolution such as authority sharing and organizational automation.

Designing new systems involves the participation of several actors and requires purposeful and socially acceptable scenarios. Scenarios are coming from stories that are incrementally categorized. They are necessary for strategic design thinking. John Carroll presents the scenario-based design approach that support human-centered design of complex systems.

Design is or must be a socially anchored activity. Complex socio-technical systems have to be developed in a participative way, i.e., realistic stakeholders have to be involved at an early stage of design, by developing the new system in actual contexts of use. Saadi Lahlou presents a series of socio-cognitive issues that a design team should take into account to design things for the real world.

Design is certainly a creative activity, but it has to be incrementally rationalized. Following up his 1998 book, Guy Boy presents a new version of the cognitive function analysis of human-machine multi-agent systems. He introduces general properties such as emerging
cognitive functions, complexity analysis, socio-cognitive stability analysis, and flexibility analysis. He insists on the fact that taking into account experience is a key factor in design, and maturity is a matter of product testing, as well as practice evolution and emergence identification.

Automated processes involve cooperation between humans and machines. Patrick Millot, Frédéric Vanderhagen and Serge Debernard present several dimensions of human-machine cooperation such as degree of automation, system complexity, and the richness and complexity of the human component. They insist on the need for a common frame of reference in cooperative activities, and on the importance of the authority concept.

David Navarre, Philippe Palanque, Célia Martinie, Marco Winckler and Sandra Steere provide an approach to human-centered design in the light of the evolution of human-computer interaction toward safety-critical systems. They emphasize the non-reliability of interactive software and its repercussions on usability engineering. They introduce the “Generic Integrated Modeling Framework” (GIMF) that includes techniques, methods and tools for model-based design of interactive real-word systems while taking into account human and system-related erroneous behavior.

Most of the chapters in this handbook deal with automation. It was important to address the various properties of human-automation interaction that support human-centered design. Amy Pritchett and Michael Feary present several useful concepts, which include authority (a recurrent issue in our modern software intensive world), representations, interface mechanisms, automation behavior and interface error states.

As already said, software is everywhere in human-machine systems nowadays. Jeffrey Bradshaw, Paul Feltovich and Matthew Johnson present a variation of human-automation interaction where automation is represented by artificial agents. The concept of artificial agent emerged from the development of semi-autonomous software in artificial intelligence. It is consistent with the concept of cognitive functions already presented.

Evaluation was for a long time the main asset of the human factors and ergonomics discipline. Jean-Marc Robert and Annemarie Lesage propose, in two chapters, a new approach of evaluation going from usability testing to the capture of user experience with interactive systems. Interaction design has become one of the main issue, and consequently activity, in the development of modern technology. Traditional design is typically done locally and integration happens after. Using a user experience approach involves holistic design, i.e., the product is taken globally from the start. This is what I call the evolution from the traditional inside-out engineering approach to the new outside-in design approach.

An example of technique and tool for measuring human factors during design is eye tracking (ET). Lucas Stefane presents the core ET research in the field of human-machine interaction, as well as the human visual system to better understand ET techniques. Real-world examples in the aeronautical domain are taken to illustrate these techniques.

Among the many factors useful to assess human-machine interaction, fatigue is certainly mostly hidden because we tend to continue working even when we are tired, and extremely tired. Philippa Gander, Curt Graeber and Greg Belenky show how the dynamics of fatigue accumulation and recovery need to be integrated into human-centered design, more specifically by introducing appropriate defenses. When operator fatigue can be expected to have an impact on safety, systems being design should be conceived as resilient to human operator fatigue in order to maintain acceptable human-machine interaction.
People performance changes with respect to their age. Anabela Simões, Marta Pereira and Mary Panou address older people’s characteristics and design requests to accommodate their needs in order to ensure efficient, safe and comfortable human-machine interactions with respect to context. They present the issue of safe mobility for older people and technological solutions with the related advantages and inherent risks.

The various effects of culture and organization influence human-machine interaction. Don Harris and Wen-Chin Lee present the influence of these effects on human errors. People do not use systems in the same way when they come from different cultures. This pragmatic aspect of human-machine interaction needs to be taken into account seriously in life-critical systems in particular.

The Francophone school of ergonomics makes the difference between prescribed task and activity (i.e., the effective task). Sometimes we believe that task analysis as a normative approach will guaranty a straight human-centered way of designing systems. This assumes that system boundaries are well-defined. In the real world this is not the case. Systems are loosely coupled in the environment where they are working. They need to absorb change and disturbance and still maintain effective relationships with the environment. Promoting this kind of reflexion, Erik Hollnagel present the diminishing relevance of human-machine interaction.

The conclusion of this handbook focuses on the shift from automation to interaction design as a new discipline that integrates human and social sciences, human-computer interaction and collaborative system engineering. For that matter, we need to have appropriate models of interaction, context and function allocation.

References


