Cognitive Function Analysis in the Design of Human and Machine Multi-Agent Systems

Guy A. Boy

Introduction

Cognitive function analysis (CFA) started to be articulated during the 1990s in the framework of the active development of cognitive engineering (Boy, 1998). This was an alternative to the "conventional" cognitive task analysis (Crandall, Klein, & Hoffman, 2006; Schraagen, Chipman, & Shalin, 2000). From the start, CFA is based both on the distinction between task, i.e., what has to be performed, and activity, i.e., what is effectively performed, and on a multi-agent approach of human-machine interaction, i.e., cognitive functions can be organized into interactive networks. Indeed, when the work place is the playground of people and software-intensive machines, a multi-agent approach needs to be taken.

Ergonomics is the science of work and the work place. Up to now, the co-adaptation of jobs and tools was done locally and the notion of human-machine interaction was thought between a human and a machine. CFA is consistent with the evolution of our industrial world that moves from local ergonomics to **global ergonomics**. Today, the problem has become more complex in the sense that the co-adaptation includes various levels of organization. Human-machine interaction needs to be thought more broadly between three types of agents: humans, technologies and organizations. A machine agent has capabilities that enable it to act on its environment in a similar way, as a human agent would do. In addition, the very notion of agent needs to be considered not only at the local level of an individual interacting with a machine, but also at the global level of an organization and even between organizations that now include sophisticated machines empowered with their own authority. An agent is also an agency of agents (Minsky, 1985), i.e., organizations themselves are agents. This is why the cognitive function paradigm was forged to support this evolution, i.e., it is both working for the local level and the global level of interaction.

A cognitive function is defined by three attributes that are its role, context of validity and a set of resources. Therefore each agent has at least a cognitive function, and more specifically a role. For example, postmen are agents belonging to a postal service organization, i.e., an agency, and their role is delivering letters. They also represent postal services, i.e., they are responsible for what they do and accountable to someone within the organization (see the chapter on authority issues in this handbook). The **role** of cognitive function is then strongly related to responsibility and accountability of the agent owning it. However, this responsibility and accountability is limited to a predefined or emerging **context**. For example, postmen are responsible for letter delivery from 8:00 AM to noon and 2:00 PM to 5:00 PM from Monday to Friday (at least in France); this is a specified definition of a temporal context. Context may take various forms. It could define space limits such as the neighborhood where the postman is responsible for delivering letters. Both of these contexts, i.e., defined and in time and space, can be predefined. However, in some situations, they can emerge from necessity, e.g., a postman is sick and his colleagues take some parts of his/her neighborhood, therefore extend their own space context and consequently extend their time context also. When such emergence is infrequent, context can be qualified being dynamic. However, when it is persistent, the agency needs to be redefined, e.g., a new postman has to be hired, or predefined time and space contexts have to be redefined. Context could be nominal, i.e., as normal situations, or off-nominal, i.e., as abnormal or emergency situations. Finally, a cognitive function empowers an agent with control to execute related assigned tasks. This control is only possible when appropriate **resources** are available to the agent. Resources can be physical or cognitive. For example, postmen have bags, bicycles or cars to carry letters; they also have cognitive resources such as pattern matching to appropriately associate the name of the street on the envelope and a street nameplate, the number of the house or apartment and finally the name of the addressee. Note that cognitive resources are cognitive function themselves. Therefore, cognitive functions have a recursive property, i.e., there are cognitive functions of cognitive functions. In addition, other agents could own cognitive resources. For example, in the case of a strike, since the number of postmen is significantly lower than usual, postmen who remain on duty have longer hours and bigger neighborhood, i.e., this off-nominal context is different from the nominal one and implies different responsibility, accountability and control. As far as control is concerned, since context is different, postmen on duty may have to delegate some of their tasks to other people, students for example. In this off-nominal context, delegation is now a cognitive resource that can be decomposed into a set of cognitive functions such as training, supervision and management. It is interesting to notice that a cognitive function that is owned by an agent can lead to a set of similar cognitive functions distributed among a set of other agents; these cognitive functions are created (trained), supervised and managed by the initial cognitive function. Consequently, carrying out a cognitive function analysis turns out to be the development of a cognitive function network over a set of agents.

Cognitive function networks are incrementally developed through the generation of cognitive function in both the resource space and the context space (Figure 1). The development of such cognitive function networks is guided by several properties of cognitive function themselves. These properties will be described in this chapter, as well as the processes that put them at work.

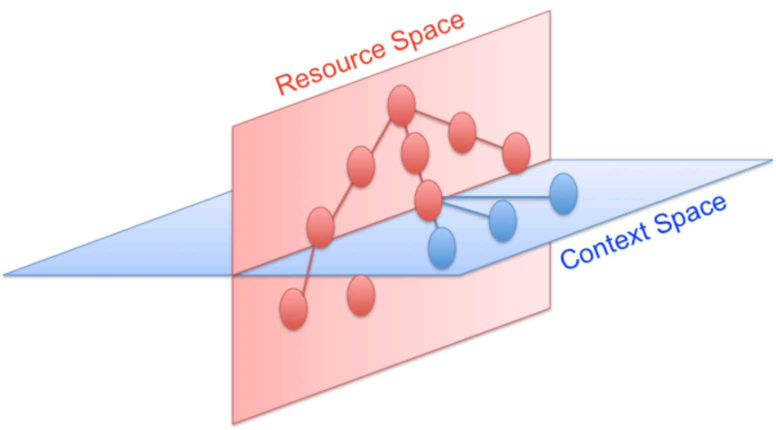

Figure 1. A cognitive function network view in both context and resource spaces.

In the resource space, configuration scenarios are described to improve the rationalization of the allocation of physical and cognitive functions to appropriate and available agents. In the context space, event scenarios (e.g., chronologies in temporal contexts) are described to improve awareness of procedural connections between cognitive functions. Consequently, a cognitive function is also defined as a process that transforms a task into an activity (Figure 2).

Figure 2. A cognitive function as a transformation of a task into an activity.

Both task (what is planned to done) and activity (what is effectively done) can be described using the same representation, called an **interaction block** (**i-Bloc**), which involves five attributes (Figure 3): a context of validity, a goal to be reached (a normal final condition of the process), a set of triggering conditions that enable to start the process, a context of validity that supersedes the triggering conditions (contextual conditions are more permanent than triggering conditions), a set of actions (typically organized into an algorithm), and a set of abnormal final conditions (in the case of failures).

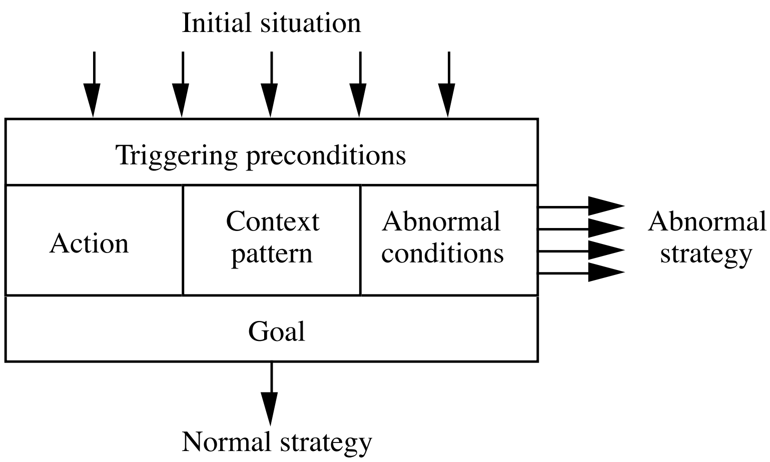

Figure 3. Interaction block representation (Boy, 1998).

Consequently, cognitive function analysis (CFA) can be carried out in two complementary ways by describing: the resource space through **declarative configuration-driven scenarios**, and the context space through **procedural event-driven scenarios**. We already saw that cognitive resources are cognitive functions themselves. Therefore, the resources space is the space that supports the elicitation of all cognitive functions in the form of cognitive function networks. In a similar way, a context of an i-Block is an i-Block itself. Consequently, i-Blocks are organized in the form of information flows by contexts, sub-contexts, sub-sub-contexts and so on. Performing a cognitive function analysis consists in articulating cognitive function networks (the resource space) and i-Blocks information flows (the context space). In the next sections, we will develop a case in order to show the applicability of CFA.

When these two representations were designed, they were not targeted toward the same objective. i-Blocks were designed to represent operational procedures. Cognitive functions were designed to represent human and machine cognitive processes. i-Blocks were designed to better understand interaction between human and machine agents. Cognitive functions were designed to better understand how a task is transformed into an activity, as well as how they can be described in terms of role, context and resources. i-Blocks are external representations of interactions, and cognitive functions are internal representations of agents' cognitive processes. To a certain extent, we could say that if i-Blocks are used to represent prescribed procedures, they are inputs of cognitive functions; in contrast, if i-Blocks are used to rationalize agents' activities, they are outputs of cognitive functions. When a cognitive function network is developed, cognitive functions are teleologically described in the resource space, and logically connected through i-Blocks in the context space.

More properties of human and machine agents' cognitive functions will be presented in the next sections of this chapter. After its initial publication (Boy, 1998), CFA has been used extensively and successfully in many research and industrial projects. Examples will be taken in the aerospace domain, but CFA is also useful in any life-critical system analysis, design and evaluation.

Properties of the resource space

There are two types of cognitive function's resources:

- cognitive resources that can be an agent or a cognitive function; and
- physical resources that are neither agents nor cognitive functions, and are typically used by a cognitive function of an agent.

For example, let's take the *Traffic alert and Collision Avoidance System* (TCAS) of current commercial aircraft. The TCAS monitors the airspace surrounding the aircraft by interrogating the transponder of other aircraft. TCAS, transponders and aircraft are represented as machine agents equipped with cognitive functions. TCAS has a first high-level cognitive function that has a role, i.e., "get information from transponders around", a context of validity, i.e., the range of the signals between TCAS and transponders (in practice the range may vary from 40 nm to 80 nm with respect to the type of TCAS), and a set of resources, i.e., software that calculates: the bearing/range to the intruder; the closure rate; the relative altitude difference and the vertical speed of the intruder (under some conditions).

If the TCAS predicts that the separation is below safe boundaries, then a *traffic advisory* (TA) message is triggered and informs the crew that the intruder is in the vicinity (this is another machine cognitive function). The crew should always attempt to visually clear the airspace before maneuvering the aircraft in response to a TA (this handled by a human cognitive function). The purpose of the TA is to advise the crew to attempt to get visual contact with the intruder. No evasive action should be solely based on the TA.

If the TCAS predicts a collision threat then a *resolution advisory* (RA) is triggered to maintain a safe separation between the aircraft. The RA is coordinated between the aircraft and the intruder; the RAs are thus complementary. The crew is then required to follow the RA promptly and smoothly. The crew should never maneuver in the opposite direction of the RA since maneuvers are coordinated. This is what the task says, but it may happen that the activity could be different such as in the Überlingen accident.

To further develop the resource space of the high-level "collision avoidance" cognitive function, that is often called the configuration space, there are three kinds of configurations that may be useful to describe:

- (Configuration-1) the two conflicting airplanes are not equipped with TCAS that are independent of *air traffic control* (ATC), i.e., traffic (collision) alert is only available to the *ATC controller* (ATCO) who typically uses radar information (R) (Figure 4);

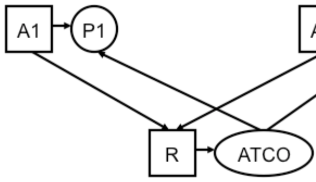

Figure 4. Configuration-1: current situation where ATCO uses radar information to anticipate possible conflicts and control separation between airplanes.

- (Configuration-2) airplanes are equipped with TCAS, and in addition, ATCO may use the ground-based *Short Term Conflict Alert* (STCA) system, which currently warns en-route ATCO when two airplanes are dangerously close to one another (Figure 5); and

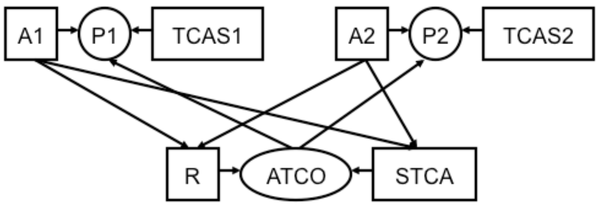

Figure 5. Configuration-2: Using TCAS onboard and STCA on the ground.

- (Configuration-3) a data-link connection exists between TCAS and STCA; this could be a future possibility (Figure 6).

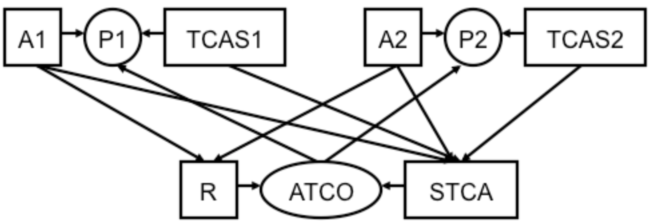

Figure 6. Configuration-3: Using TCAS connected to STCA.

What is the configuration space? In Configuration-1, main agents are the pilot (P1) of airplane 1 (A1), the pilot (P2) of airplane 2 (A2) ATCO and STCA. In Configuration-2, there are the same agents, as well as TCAS1 for A1 and TCAS2 for A2. In Configuration-3 that does not exist today, TCAS and STCA are able to talk between each other.

Properties of the context space

Continuing on the collision avoidance problem, let's assume that two airplanes converge toward each other. As already described in the resource space, below a specific separation distance, there should be a traffic alert that triggers an abnormal situation. This is interpreted in terms of cognitive functions as follows: the set of cognitive functions belonging to a "normal" context are replaced by the set of cognitive functions belonging to the "traffic alert" context. In this new context, the highest cognitive function's role, or goal, is to go back to a normal context.

In Configuration-1, ATCO is the only agent to have the authority to separate A1 and A2. ATCO manages the separation using radar information and sends requests to P1 and P2 according to the situation, i.e., whenever a conflict is detected, ATCO is in charge. Therefore, in terms of i-Blocks, the first triggering condition is the detection by ATCO of a possible conflict; the goal is to change A1 and/or A2 trajectories in order to solve this conflict. ATCO sends requests to pilots, e.g., "climb" to P1 and "descent" to P2. Possible abnormal conditions are, for example, radio transmission failures and pilot inattention to radio messages. On the pilot side, actions have to be taken (another i-Block) that leads to another i-Block on ATCO side for the monitoring and acknowledgement of appropriate pilots actions and finally effective conflict resolution. The i-Block network is pretty simple, and provides a very clear explanation of ATCO-centered authority.

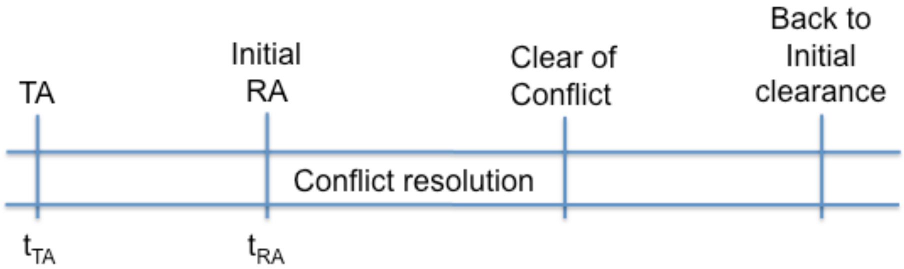

Figure 7. Timeline of TCAS alert and conflict resolution.

In Configuration-2, each TCAS monitors the airspace around the airplane for another TCAS transponder. It warns the pilot when it detects the presence of another TCAS. TCAS sends a traffic alert (TA) requesting the pilot to "climb" or "descend". Then, the pilot sends to ATCO a resolution advisory (RA) after a period of time $t_{RA} - t_{TA}$ (Figure 7). After the conflict is cleared, ATCO sends a "clear of conflict notification". The main problem comes from the fact that during $t_{RA} - t_{TA}$, ATCO, who does not know about TCAS TA, could possibly send conflicting requests to P1 and/or P2. This is clearly explained by the fact that pilots' and ATCO's conflict resolution i-Blocks are not connected. This issue is solved in Configuration-3, i.e., ATCO knows about both TCAS traffic alerts. If this kind of analysis would have been performed at design time, the probability of accidents such as the mid-air collision that occurred at Überlingen in 2002 would have been significantly decreased. In particular, abnormal conditions could be generated only by analysis, i.e., the development of such analytical scenarios could generate the emergence of such absence of links between crucial i-

Blocks. In addition, the use of i-Block networks linking agents between each other in a dynamic way affords to figure out the way displays and controls should be designed. Therefore, in this specific aviation case, “collision avoidance” displays provided to both pilots and ATCOs should be coordinated. The link between TCAS and STCA for example is a new resource and generates new contexts shared by both pilots and ATCOs.

By definition, a context of i-Blocks is an i-Block itself (see Figure 8). Therefore, i-Block construction may be done alternatively from inside-out or outside-in, i.e., developing bottom-up and top-down task analysis. The former attempts to induce generic i-Blocks from experience data; the latter attempts to decompose high-level i-Blocks into more specific i-Blocks. In practice, both approaches are used to mutually bootstrap each other.

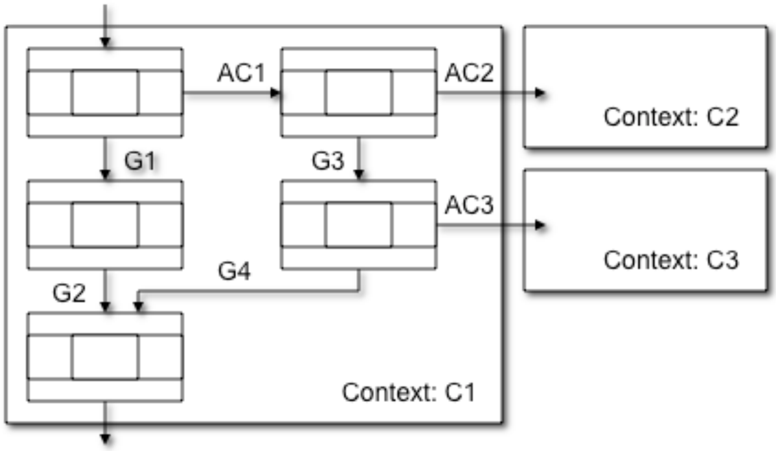

Figure 8. Contexts of i-Blocks.

Consequently, developing an i-Block network is like “making a house with prefabricated blocks”. Note that this construction, typically made in the context space, should be carried out concomitantly with the construction of the related cognitive function network in the resource space. Let’s concentrate on this analogy. In order to get a beautiful house, an architect is strongly needed in the first place to provide professional directions based on experience and knowledge. This is the same for the design and development of an i-Block network, experience and knowledge of an i-Block architect is required. This type of architect is called a *cognitive engineer*. Obviously, cognitive engineers base their activity on domain actors, e.g., railway engineers and train drivers or aerospace engineers and pilots. They know for example that the more an i-Block network is linear, i.e., a chain of i-Blocks going from one to another on a normal strategy only, is better than an i-Block network that would have too many transversal abnormal strategy branches. They also know that a too linear i-Block network is a “theory” that would break when put into the real world. Therefore, i-Block networks are able to suggest what kind of experiments should be performed to elicit relevant normal or abnormal strategies.

We already introduced the concepts of normal, or nominal, and abnormal, or off-nominal, contexts, strategies and i-Blocks. Normality is a notion that can be static, i.e., domain-dependent, and/or dynamic, i.e., context-dependent. But even in the former notion, the domain may evolve, normality is sensitive to technology maturity as well as maturity of practice. This issue of maturity will be further discussed later in this chapter. The latter notion of normality is adaptable according to context. This adaptable normality concept is essential to better understand the distinction between weak and strong abnormal conditions. A weak abnormal condition (e.g., AC1 in figure 8) leads to a set of i-Blocks that have the same context pattern, i.e., the generated abnormal strategy stays within the same context. A strong abnormal condition (e.g., AC2 and AC3 in figure 8) leads to a set of i-Blocks that do not have the same context patterns, i.e., the generated abnormal strategy does not stay within the same context of the initial i-Block.

General properties

Emerging cognitive functions

A cognitive function can be defined a priori; in this case, it is a **deliberate** cognitive function. Deliberate cognitive functions can be implemented in a machine, usually in the form of software, or learned by a human being. However, it may happen that a new technology or the way work is organized among different interacting agents induces the **emergence** of new relevant cognitive function, i.e., cognitive functions that are not defined a priori. For example, text-processing tools that we use today have two important functions, i.e., copy and paste, which emerged from the technological possibilities provided by graphical displays and interactive computers. These functions were not specifically developed to support text-processing tools in the first place. Text-processing tools were designed to enable people to write text without needing an external resource, such as a secretary, to type written reports, papers or books. However, the copy and paste functions naturally emerged because it was easy to implement them using graphical displays and interactive technologies such as the mouse. These functions are both cognitive and physical because their resources are both cognitive and physical. On the cognitive side, we shifted from linear writing to non-linear writing. Indeed, in the past, we wrote manually using paper and pencil; our thinking was forced to be linear. This is why we learned in primary school to design an outline of the text, and then we were writing chapters, paragraphs and sentences linearly. Today, we have several tools that enable us to write in a non-linear way an introduction, a conclusion and everything that is in between. It is possible to backtrack as many times as we want on what has been already written and modify it easily. We use "copy", "cut" and "paste" machine cognitive functions… In other words, the global cognitive function of linear writing has been totally modified into a global emerging cognitive function of non-linear writing. This took a decade or so to see the emergence of this new cognitive function, and stabilize the related maturity of practice.

An emergent cognitive function, or emerging cognitive function, can be implemented within a technology, e.g., "copy" and "paste" within text processing and beyond in desktop applications. It can also stay at the level of practice and anchor itself into the organizational environment. Office automation emerged as a technology and a practice; it contributed to remove secretary's traditional functions, and in many cases secretaries themselves.

Therefore, an emergent cognitive function is a cognitive function that naturally emerges from the use of a new tool or the implication of an agent into a new organization. An emergent cognitive function can become mature in a variable amount of time. This time is very difficult to predict. For example, MP3 uploading mainly emerged from the use of iPods even if we believed that CDs were there to stay as the major music support. The notion of cognitive function is thus related to the notion of practice, like the practice of CDs or MP3. In general, a cognitive function emerges because resources that are associated to it are easier to implement and use, and also because the context of this implementation is appropriate. Note that fashion phenomena could also drive this emergence. The obsolescence or persistence of the resulting cognitive function is then a question of robustness and resistance to new competing emergent cognitive functions.

Complexity analysis

In multi-agent systems, complexity mainly results from the multiple links between agents. Highly connected multi-agent systems behave like biological systems. Many definition of complexity may be taken to assess complexity. Complexity can be expressed in terms of the number of (possibly interconnected) problems to be solved at a given time for a given human agent. For an ATCO, complexity can be expressed as the number of relevant aircraft to be managed per appropriate volumetric zone (AVZ) at each time. An AVZ is calculated with respect to the type of flow pattern, e.g., aircraft crossing, spacing and merging. The definition of such an appropriate volumetric zone requires the assistance of operational ATC controllers. From a socio-cognitive perspective in ATM, complexity should be considered together with capacity. This is what the COCA (COmplexity & CApacity) project investigated (Athènes et al., 2002; Cummings & Tsonis, 2006; Hilburn, 2004; Laudeman et al., 1998; Masalonis et al., 2003). This can be expressed in terms of cognitive functions and available resources in context. When the airspace capacity increases for example, the number of resources should also increase. This is why it is crucial to carry out a cognitive function analysis to investigate the relevance and appropriateness of cognitive function allocation to software. Criteria, such as workload and situation awareness acceptability, should be defined to assess such allocation. In any case, it is always useful and often mandatory to run human-in-the-loop simulations (HITLS) in order to further elicit, validate and verify cognitive function networks. This complexity analysis is carried out concurrently in both resource and context spaces.

Socio-cognitive stability analysis

In highly connected multi agent systems, such as our evolving airspace with a constant growth of interdependencies among agents, it is necessary to be able to assess socio-cognitive stability (SCS). SCS can be derived from various contributions, including Latour's account on socio-technical stability (Callon, 1991; Latour 1987), emerging cognitive functions (Boy, 1998), distributed cognition (Hutchins, 1995), and socio-cognitive research and engineering (Hemingway, 1999; Sharples et al., 2002). In previous work, we made a distinction between local and global SCS (Boy & Grote, 2009). Local SCS is related to agent's workload, situation awareness, ability to make appropriate decisions and, finally, correct action execution. It can be supported by appropriate redundancies and various kinds of cognitive support such as trends, relevant situational information and possible actions, i.e., cognitive function resources. Global SCS is concerned with the appropriateness of functions allocated to agents, pace of information flows and related coordination, i.e., an improved understanding of the context space. It is very similar to the level of synchronization of rhythms in a symphony. Globally, socio-cognitive support could be found in a safety net that would take into account the evolution of interacting agents and propose a constraining safety envelope in real time.

Flexibility analysis

Flexibility is defined as the ease of modification of a contract between two or several agents in real-time, e.g., an air-ground contract that specifies the trajectory of an aircraft. Flexibility assessments should guide cognitive function allocation in both resource and context spaces, e.g., in the ATM of the future, resource allocation mainly means human-centered automation and context definition; and management means organizational setting. As already said, increasing capacity also increases complexity and uncertainty, which need to be managed by finding the right balance between reducing uncertainties through centralized planning, i.e., compiling cognitive functions and their interconnections, and coping with uncertainties through decentralized action. The main problem comes from the fact that the more we reduce uncertainty by planning, the more flexibility becomes an issue. Loose coupling is required for actors to use their autonomy in accordance with system goals (Grote, 2004).

Activity analysis and human-centered design

Understanding and taking into account people experience in design

Human factors and more specifically cognitive engineering have been recently concentrating on taking into account human issues in design and development of both technology and emerging societal practices. *Human-centered design* (HCD) is about designing technology tailored to people. I prefer to talk about people and not about users because it is a broader concept, even if in many cases we will still use the term "user". Users deal with tools. Users are often customers. People deal with life. This is why the concept of life-critical systems is crucial here.

Technology has multiple facets; it can be used, maintained, repaired and eventually dismantled. In other words, there will be people who will use, maintain, repair or dismantle technology. This is true for houses, cars, airplanes and nuclear power plants, for example. A main objective is to design and develop for people who are not immediately perceived as obvious customers. Therefore, when we talk about human-centered design, we need to be careful to focus on all people who will interact with the technology being produced, as much as possible. Consequently, cognitive functions that we will be trying to elicit and rationalize will concern all possible agents dealing with the product being designed.

On the one hand, if too much emphasis on current practice should not guide the design of a novel interface, it is crucial to understand the constraints and requirements that people have when they perform their work now. It is important to understand why they cannot accomplish their work properly or, conversely, perform it very well in a wide variety of situations. Both positive information and negative information on work practice are equally good to consider and analyze. People experience cannot be separated from the tools, methods and organizational setups that go with it. The difficult part is to access the right people, and not intermediary people who would synthesize their views on what should be the requirements for the design team. Obviously, all users cannot be accessed in all possible situations. However, by experience, selecting appropriate sets of users is much better than nothing! We need to remember that resulting acquired information is partial. This is why we need to have conceptual models that support interpretation and extrapolation in some cases. These conceptual models may be very loose and provided by domain experts in the form of narratives or simply active explanations of acquired information. Cognitive functions, as a mediating representation between human operators and designers, provide such a model.

On the other hand, it is possible to analyze some parts of possible future people experience from experts, but there are situations, configurations or initiatives that will never be possible to anticipate and therefore only a prototype-based approach will enable the elicitation of possible operational patterns. First, there are behaviors that are standard and could be anticipated because they are related to a style of interface, for example. The more the interface conforms a standard, the more behaviors will be predictable. Standardization is therefore a great incentive for future people experience prediction. Nevertheless, when new kinds of systems and user interfaces are designed and developed, usability predictability is no longer possible without an experimental protocol that involves a set of users facing a prototype.

This is why both current-activity analysis (present) and emerging-activity analysis (future) require a mediating representation in the form of cognitive functions. The former is based on the observation of what is being done today, and the latter is based on human-in-the-loop simulations and related activity observation and analysis. As a matter of fact, activity analysis should be a constant effort carried out during the life cycle of a life-critical product.

Designing for maturity

When a Chef is cooking a memorable dish, his or her experience and expertise play a crucial role, and incremental testing is a must. We do not insist enough on this necessary capacity of domain experts to be involved and concentrated during the design process. We often talk about latent human errors (Reason, 1990) that are committed during the design and development process and re-appear at use time, sometimes viciously. If we take a positive approach of this problem, latent errors deal with product maturity. Product maturity is a matter of constant testing by experts. In an ideal world, we would have to test the product and its former prototypes in all situations in order to make sure that it will be fully mature at delivery time. This is obviously impossible, but designers and engineers must remember that the more situations they will experience in the use of the product, the better. This is a practice that happens to disappear with our current industrial way of managing projects. Indeed, engineers need to fill in spreadsheets and report all the time instead of fully concentrating on their design and development tasks. It seems that reporting has become more important than actual design and development! Motivation must be kept and creativity must remain the main asset of human-centered design teams. For that matter, reporting could be used in a different way that would effectively and significantly improve design. At this stage, it is important to make a distinction between reporting for work FTE (full-time equivalent) justification and writing for improving design.

A technology becomes mature when it is useful, usable and acceptable, i.e., is socially accepted, meets legal requirements, and answers relevant commercial issues. Appropriate people experience must be taken into account to enhance maturity in engineering approaches. All actors involved in the life cycle of the product, e.g., end-users, customers, maintainers, trainers, and designers, must be taken into account, as well as the repercussions on other people of its use and eventually the deconstruction of the product. In the early stages of a technology, products are driven by the needs of technically sophisticated consumers, but these needs should be reevaluated when the technology matures.

The operational life cycle of a technology can be divided into two periods that are characterized by different maturity criteria: (period 1) technology and performance; (period 2) ease of use, reliability and price. We need to take into account that people behavior changes when they are using such technology. For example, in the early stages of computer industry development in the 70s, computers were big and mostly used by highly-skilled engineers. Computer use was a matter of technical performance. Microcomputers emerged and democratized the use of information technology to the point that most people have a computer at home today. Microcomputers arrived during the first half of the 80s. Many engineers at that time did not want to use such a new technology because they thought that it was made for technically low-skilled users. The transition from period 1 to period 2 was being reached at that point. Today, computers are becoming invisible, integrated within the most familiar tools such as the telephone, automobile, or microwave. This is another transition point. We will say cognitive functions changed.

At such transition points, maturity is an issue. The best way to master maturity is to improve the period 0 that includes design and development of the product. The main issue here is that it is very difficult and almost impossible to predict the future without relevant data. Experience feedback and expert knowledge are often required to make appropriate design and development decisions. Instead of periods 0, 1 and 2, it is much better to work on periods n, n+1 and n+2. This assumes that we work on a family of products. This product family issue is crucial and has emerged for many industrial products including aerospace, software and telecommunication. Thus n-1 knowledge is incrementally used in period n. The incremental development of cognitive function networks is an important support for understanding and managing various human-machine interactions, and therefore deciding potential changes in either technology or people practices.

CFA-based design

Design and development are typically organized top-down. Everything starts with an idea, e.g., building a new aircraft that will be able to transport 800 people. Then, technical experts meet to examine this very high level requirement and the current technological possibilities. Usually, there are various kinds of technological innovations that may need to be experimented to develop an appropriate solution. However, the initial top-down goal-driven approach must be cross-fertilized by an event-driven approach also, i.e., solutions must be incrementally tested on appropriate scenarios. From a human factors point of view, scenarios are key in a human-centered design process. They are difficult to develop and I advise the reader to refer to the chapter on scenario-based design included in this handbook. The development of scenarios and their use in human-in-the-loop simulations are a good way to elicit cognitive functions involved in the interactions between humans and technology. In the introduction of this volume, I introduced the AUTOS pyramid as a conceptual tool to guide human-centered design. AUTOS supports the analysis and synthesis of cognitive functions into five generic categories related to Artifacts (i.e., the technology being designed and developed), Users (i.e., people who will interact one way or another with this technology), Tasks (i.e., things that needs to be accomplished using this technology), Organizations (i.e., the various ways people are interconnected) and Situations (i.e., environment status and events).

There are many ways to describe agents' activities. In this chapter, I will take two distinctive categories, declarative and procedural descriptions. These distinctions are not new and were used in artificial intelligence for a long time. For example, a cognitive function can be represented in a declarative way by specifying three attributes: a role, a context of validity, and a set of resources; and i-Blocks support the procedural description of information flows between agents and more specifically cognitive functions.

Discussion

CFA is one of the many approaches useful for analyzing, designing and evaluating interactive systems in a human-centered way. Unlike previous approaches that were based on a single-agent human operator model (Rasmussen, 1983; Endsley, 1988; Vicente & Rasmussen, 1992), CFA straightaway provides a very usable and useful framework for socio-technical and organizational analysis, because it is multi-agent by construction.

Cognitive work analysis (CWA) for example is commonly based on domains that are incrementally redesigned (Vicente, 1999). CWA typically starts with an analysis of social and organization factors but does not have any multi-agent formalism to support the description of such analysis. The first formal step of CWA is the work domain analysis (WDA) that consists in the identification of all the goals and purposes of the system being studied. WDA is based on data, information and knowledge from existing documentation and expert elicitation results. WDA is performed using either the abstraction-decomposition method (Rasmussen, 1985) or the abstraction hierarchy method. The former method includes a sequence of five steps that consist in determining goals, priority measures, general functions, processes and objects. The latter method is very similar but attempts to answer questions such as "Why" and "How" for each provided piece of information. The second formal step of CWA is the control task analysis, which involves the identification of the control tasks that are performed within the system being analyzed. A control task analysis is used to determine what tasks are undertaken within the system being analyzed, regardless of how they are undertaken or who undertakes them. Decision ladders are used for the control task analysis component of CWA. The third step consists in analyzing strategies. This phase involves identifying and representing the strategies that actors within the system being analyzed use when they perform the control task identified during the control task analysis phase. Information flow maps are used for the strategies analysis component of CWA. In the fourth step, we could see the social organization and co-operation analysis of CWA as the identification of how control tasks are allocated among agents and artifacts within the system. It uses the abstraction-decomposition hierarchy, decision ladders and information flow maps previously developed, but does not involve any specific modeling tool. The last step involves the identification of the cognitive skills required for the performance of the control task. Rasmussen's *Skill, Rule, Knowledge* (SRK) framework is used to categorize these activities.

Another approach is the *Goal-Directed Task Analysis* (GDTA) that focuses on the production of situation awareness requirements (Endsley et al., 2003). Situation awareness is typically decomposed into three levels: (1) perception of elements in the environment; (2) comprehension of the current situation; and (3) projection of future status (Endsley, 1988). GDTA attempts to break down a specific work domain into various goals and subgoals that have specific SA requirements that can be elicited. First a goal hierarchy is produced from previous appropriate knowledge and information. This hierarchy is typically limited to three levels. Like in CFA, GDTA requires that the goal or role of each function should be explained. Then a set of secondary interviews is carried out in order to validate the first elicited hierarchy. The resulting structure is then used to generate appropriate questions that address situation awareness requirements. Finally, feedback is conducted to validate the resulting GDTA structure until a high level of consistency, coherence, and completeness has been achieved.

More recently, a new technique that combines CWA and GDTA was proposed (Humphrey & Adams, 2009), the *Cognitive Information Flow Analysis* (CIFA). The authors put to the front that GDTA focuses on part-whole relationships, as CIFA and CWA focus on producer-consumer relationships. This distinction is essential. The part-whole relationship denotes the teleological nature of functions. The producer-consumer relationship denotes the logical nature of functions. In CFA, there are two planes of interconnectivity, the teleological one (i.e., using part-whole relationships that connect functions between each other) and the logical one (i.e., using producer-consumer relationships that connect i-Blocks from one to another). In addition, the notion of function in CFA is somehow different than in CIFA because CFA takes into account the concept of role, and therefore authority, as a basic attribute. CFA intrinsically includes the notion of control and accountability (and responsibility).

What these other approaches do not include at all is the notion of context. Context is crucial because it is the unifying link between the teleological nature and logical nature of cognitive functions. The co-development of cognitive function networks and i-Block networks results in chains of accountability in the sense of functional and operational traceability and reliability. In addition, these networks provide a very good framework for verification and validation, and more importantly test the degree of maturity of both technology and practice.

Furthermore, CFA is a system-level framework that differs from a user-interface-level framework such as GOMS (Card, Moran & Newell, 1983; Irving et al., 1994). CFA make a distinction between task-content-related cognitive function and interaction-related cognitive functions (Boy, 1997). The part-whole nature of CFA provides the analyst with enough relief to represent the task content part of a high-level function and the interaction part of a low-level function. In a sense, when it comes to describe interaction cognitive functions, CFA cognitive functions could be modeled as GOMS methods.

The more agents are interconnected, the more the resulting multi-agent systems is complex, and the more it is difficult to isolate part to study them locally. The separability issue imposes a global approach of multi-agent systems. This is why both resource and context spaces need to be considered and developed in concert. Both local and global socio-cognitive stability need to be assessed to figure out where real problems are. If analytical studies using CFA are useful to start a design process of a multi-agent system, HITLS are necessary to observe and discover emerging cognitive functions and further rationalize resulting cognitive function networks. Consequently, CFA needs to be used incrementally during the life cycle of a multi-agent system. This leads to the maturity issue that was presented earlier in the chapter.

Conclusion and research perspectives

This chapter presents an evolution of the cognitive function analysis from the perspective of developing both resource and context spaces using the properties of both cognitive functions and i-Blocks. Contexts are difficult to define a priori during design unless experience and expertise is heavily used. In fact, they emerge from incremental refinement of cognitive function and i-Block networks first generated analytically and eventually refined experimentally from the results of human-in-the-loop simulations (HITLS). Understanding contexts of use is crucial in order to maximize the anticipation of possible surprises. It is then important to elicit persistent operational patterns to reduce uncertainty during subsequent operations. However, it is also important to be careful to understand the product resulting from automation, i.e., the integration of these patterns into software and systems, and the emergent cognitive functions resulting from this automation. Cognitive function and i-Block representations are useful to support both the discovery of progressive emergence and consolidation of generic contexts. The development of cognitive function and i-Block networks helps in the rationalization of the design and evaluation of multi-agent systems by better understanding the various assigned roles in appropriate contexts of operations with the provision of relevant and available resources.

Summary

The cognitive function representation is used to describe the role of a human, a machine or an organization in a given context, involving a set of resources. *Cognitive function analysis* (CFA) is an approach and a method that enables the description of cognitive function allocation among human and machine agents in highly-automated life-critical systems. It consists in incrementally describing both intentionally-created, as well as emerging, cognitive functions. CFA can be typically carried out from the first idea during the design phase to the obsolescence of the system itself. Cognitive functions are elicited at the same time as interaction blocks (i-Blocks) that links them between each other. Each i-Block is described in a procedural way by specifying a quintuplet (goal, triggering conditions, context, actions and abnormal conditions). Both cognitive function and i-Block formalisms enable the description of declarative configuration-driven scenarios and procedural event-driven scenarios, which guide formative evaluations of the system being designed.

References

Athènes, S., Averty, P., Puechmorel, S., Delahaye, D., & Collet, C. (2002). ATC complexity and Controller Workload: Trying to bridge the gap. *Proceedings of HCI-Aero'02*, J. Hansman, S. Chatty & G. Boy (Eds.), Boston, USA.

Boy, G.A. (1997). Cognitive function analysis: an example of human-centered re-design of a flight management system. *Proceedings of the 13th Triennial Congress of the International Ergonomics Association*, June 29 -July 4, Tampere, Finland.

Boy, G.A. (1998). *Cognitive function analysis*. Ablex. Distributed by Greenwood Publishing Group, Westport, CT, USA. ISBN 1567503764, 9781567503760.

Boy, G.A. & Ferro, D. (2003). Using Cognitive Function Analysis to Prevent Controlled Flight Into Terrain. Chapter of the Human Factors and Flight Deck Design Book. Don Harris (Ed.), Ashgate, UK.

Callon, M., (1991), Techno-economic networks and irreversibility, in Law, J., (Eds.), *A sociology of monsters: essays on power, technology and domination,* Routledge, London, pp. 132-161.

Crandall, B., Klein, G. & Hoffman, R.R. (2006). *Working Minds: A Practitioner's Guide to Cognitive Task Analysis*, Bradford Book/MIT Press, Cambridge, MA, USA.

Cummings, M. L. & Tsonis, C.G. (2006). Partitioning Complexity in Air Traffic Management Tasks. *International Journal of Aviation Psychology*, Volume 16, Issue 3 July 2006, pp. 277–295.

Endsley, M. (1988). Design and evaluation for situation awareness enhancement. *Proceedings of the Human Factors Society 32nd Annual Meeting*. Human Factors Society, Santa Monica. pp. 97-101.

Endsley, M., Bolté, B. & Jones, D. (2003). *Designing for situation awareness: An approach to user-centered design*. Taylor and Francis, New York.

Grote, G. (2004). Uncertainty management at the core of system design. *Annual Reviews in Control*, 28, pp. 267-274.

Hemingway, C.J. (1999). Toward a Socio-cognitive Theory of Information Systems: An Analysis of Key Philosophical and Conceptual Issues, *IFIP WG 8.2 and 8.6 Joint Working Conference on Information Systems: Current Issues and Future Changes*. Helsinki, Finland: IFIP, pp. 275-286.

Hilburn, B. (2004). Cognitive complexity in air traffic control : A literature review. Project COCA - COmplexity and CApacity. EEC Note No. 04/04.

Hutchins, E. (1995). How a Cockpit Remembers its Speeds. *Cognitive Science*, 19, pp. 265-288.

Irving, S., Polson, P. & Irving, J.E. (1994). A GOMS analysis of the advanced automated cockpit. *Proceedings of CHI'94*. ACM Press. Boston, MA.

Latour, B., (1987). *Science in action: how to follow scientists and engineers through society.* Harvard University Press, Cambridge, MA.

Laudeman, I. V., Shelden, S. G., Branstrom, R. & Brasil, C.L. (1998). Dynamic Density. An Air Traffic Management Metric. California: National Aeronautics and Space Administration, Ames Research Center.

Masalonis, A. J., Callaham, M. B., & Wanke, C.R. (2003). Dynamic Density and Complexity Metrics for Realtime Traffic Flow Management. Presented at the ATM 2003 Conference, 5th EUROCONTROL/FAA ATM R&D Seminar (Budapest, Hungary).

Midkiff, A.H., R.J. Hansman & T.G. Reynolds (2004). Air carrier flight operations. Report No. ICAT-2004-3. MIT International Center for Air Transportation. Department of Aeronautics and Astronautics, MIT, Cambridge, MA, USA.

Rasmussen, J. (1983). Skills, Rules and Knowledge: Signals, Signs and Symbols, and other distinctions in human performance models. *IEEE Transactions on Systems, Man and Cybernetics*, vol. 13, pp. 257-266.

Rasmussen, J. (1985). The role of hierarchical knowledge representation in decision making and system management. *IEEE Transactions on Systems, Man and Cybernetics*, vol. 15, pp. 234-243.

Sharples, M., Jeffery, N., du Boulay, J.B.H., Teather, D., Teather, B. & du Boulay, G.H. (2002). Socio-cognitive engineering : A methodology for the design of human-centered technology. *European Journal of Operational Research*, Volume 136, Issue 2, January, pp. 310-323.

Schraagen, J.M., Chipman, S.F., & Shalin, V.L. (Eds.). (2000). *Cognitive task analysis*. Lawrence Erlbaum Associate, Mahwah, NJ, USA.

Vicente, K. (1999). *Cognitive Work Analysis. Toward safe, productive and Healthy Computer-based Work*. Mahwah, Erlbaum, NJ.

Vicente, K. & Rasmussen, J. (1992). Ecological interface design: Theoretical foundations. *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, pp. 589-606.