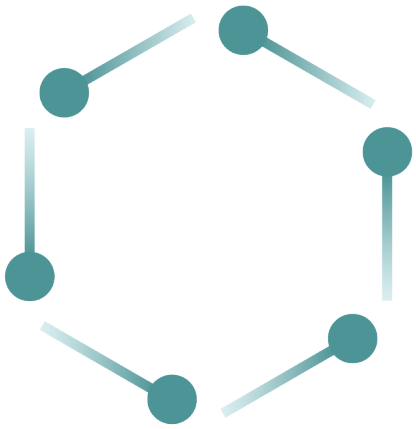


D2.1 – Design principles for digital assistants and HF assessment methodology

Document number	D2.1
Document title	Design principles for digital assistants and HF assessment methododology
Version	**1.0**
Work package	WP 2
Edition date	30.09.2023
Responsible unit	RISE
Dissemination level	PU
Project acronym	SafeTeam
Grant	101069877
Call	Safe, Resilient Transport and Smart Mobility services for passengers and goods (HORIZON-CL5-2021-D6-01)
Topic	HORIZON-CL5-2021-D6-01-13: Safe automation and human factors in aviation – intelligent integration and assistance

This project has been funded by the European Union under Grant Agreement 101069877

SafeTeam Consortium

innaxis	Innaxis (INX)
AESA AGENCIA ESTATAL DE SEGURIDAD AÉREA	Agencia Estatal de Seguridad Aérea (AESA)
TUM	Technische Universität München (TUM)
DataBeacon	DataBeacon
ONERA THE FRENCH AEROSPACE LAB	ONERA
RI. SE	Rise Research Institutes of Sweden AB (RISE)
PEGASUS AIRLINES	PEGASUS HAVA TASIMACILIGI ANONIM SIRKETI (PEGASUS)
UK Civil Aviation Authority International	CAA INTERNATIONAL LIMITED (CAAi)
L3HARRIS™	L3HARRIS FLIGHT DATA SERVICESLIMITED

Document change record

Version	Date	Status
1.0	30/09/2023	Final document ready for submission

Abstract

The SafeTeam project aims to investigate the power of digital assistants and how new technique can improve safety in the aviation domain and incorporating human factors theory to ensure that safety measures are followed throughout the implementation process. The work presented in this deliverable describes the development of a framework with the purpose to aid individuals who lack expertise in human factors to consider such aspects and improve human-autonomy collaboration.

The SafeTeam framework has been developed in parallel with two of the project's use cases, which made iterative refinement based on feedback from those processes possible. The resultant framework unfolds in three different phases. Employing Hierarchical Task analysis as a guiding tool, it navigates the reader through critical stages: reflecting on the existing system and associated tasks, designing the function allocation for optimal human-autonomy teaming, and assessing risks of the proposed design. Completing these three phases, readers will emerge with a comprehensive set of design considerations that will serve as a resource for the subsequent development process.

Table of Contents

Abbreviations

CSE	Cognitive systems engineering
CTA	Cognitive task analysis
HAI	Human-automation interaction
HAT	Human-autonomy teaming
HF	Human factors
HMI	Human Machine Interface
HTA	Hierarchical task analysis
JCF	Joint control framework
LOA	Level of automation
ML	Machine Learning
OOTL	Out-of-the-loop
OPD	Observability, predictability and directability
PFD	Primary Flight Display
SA	Situation awareness
SoA	Sense of agency
TTA	Tabular Task Analysis

1 Introduction

The introduction of AI-based tools into a system is likely to profoundly alter the nature of interactions between humans and these systems. In particular, the addition of AI tools requires that human operators in charge of these systems be able to coordinate effectively with these emerging assistance systems. Such coordination constraints need to be considered as early as possible in the design cycle. The SafeTeam project aims to **provide a methodology for considering and compensating for these new *Human Factors* (HF) constraints and for improving man autonomy collaboration**.

The methodological approach presented in this report is designed for researchers and practitioners who have no or limited experience with Human Factors and Human-automation interaction. People or organizations with the resources and expertise in these areas should utilize their existing knowledge and more complete methods to achieve the desired results. Instead, the SafeTeam method was inspired by the Pareto principle; an organization could use the proposed method to "achieve 80% of the results for 20% of the cost." This means that the framework is deliberately simplified to provide novice practitioners with a tool that is easy to use while taking the most relevant Human Factors and Human-Automation Interaction (HAI) issues into account without requiring labor-intensive processes or intimate knowledge of the relevant literature. The method does not go *in-depth* into the areas it covers. Rather, it is designed to *widen* one's perspective to dimensions which were not previously considered. In this way, the method does not necessarily replace development process tools (design, implementation, validation, etc.) utilized by industry entities, but serves to complement these by offering different perspectives and potential insights.

The method is mainly intended to support further development or iteration upon an existing system. Applying it to design of an entirely novel system may require additional modifications.

The SafeTeam design framework will not, on its own, ensure aviation safety. Suppose that safety consists of (at least) three dimensions; technical safety, organizational safety, and cognitive safety. These can map approximately onto what is sometimes called the TOP model, or (T)echnology, (O)rganization, and (P)eople. The SafeTeam design framework focuses mainly (but not exclusively) on the cognitive (or People) aspects of safety, see Figure 1. Design outcomes from this process must therefore be evaluated alongside Technology and Organizational requirements and limitations. Optimally, this would be done *during* design to allow for rapid redesign (e.g., due to technology limitations), rather than *after* a design phase has concluded a final design (at which point it might be too late or expensive to scrap a design).

Figure 1: Safety as generated and maintained by technology, organization, and people. The SafeTeam method primarily deals with people. Depending on the specific use case, the method may also contribute designs or requirements related to technology or organization.

Concretely, the proposed methodology identifies the new coordination constraints for the human operator generated by the introduction of an assistance algorithm endowed with a certain level of autonomy. It also identifies ways of compensating for these constraints through the design of more cooperative artificial agents. The objective of this methodology is therefore limited to the problem of cooperation between human operators and artificial agents. On the other hand, it does not solve all the HF issues linked to human-machine interface (HMI) design.

2 Theoretical background

This chapter will delve into the theoretical background that constitutes the foundation of the work performed in work package 2.1. The fundamental principles of automation and autonomous systems and the challenge of distributing tasks between humans and machines will be explored alongside a range of techniques and methods aimed at optimizing human-autonomy teaming.

2.1 Automation and Automatic vs. Autonomous Systems

Humans have been inventing ways to automate their work for thousands of years, from the mechanical ingenuity of ancient Greece, through steam engine regulation and control during the industrial revolution, to the introduction of electric control relays and modern computers. As such, human control activities have shifted from being physical in nature, to cognitive. Consequently, skill requirements have also shifted from the perceptual-motor skills needed for manual work, to the cognitive skills (e.g., planning, monitoring, decision-making, etc.) required to supervise automated processes (Woods, 1985). Indeed, *automation* is typically — but not exclusively — defined as the process of introducing physical or digital equipment and devices (*artifacts;* Norman, 1993) to replace human process operation or intervention, partially or fully (Janssen et al., 2019; Kaber, 2018). An *automated* or *automatic* system, therefore, is a system designed to repeatedly perform pre-defined and deterministic actions in sequence to achieve specific outcomes (Hancock, 2019). A related but distinct concept, *autonomy*, refers to the capability of an *agent* to independently select, initiate, and control its observation, decision, and action behavior in its environment (Deci & Ryan, 1987; Johnson et al., 2011a; Kaber, 2018). An *agent*, in this sense, is a biological, mechanical, or digital system that relies on sensors (e.g., eyes, cameras, etc.) and actuators (e.g., hands, motors, etc.) to perceive and act within its environment (Janssen et al., 2019; Kaber, 2018; Russell & Norvig, 2014). Thus, an *autonomous system* is an agent that does not rely on externally provided knowledge (Russell & Norvig, 2014) but learns and evolves its behavioral capacity by integrating input feedback and contextual information into its knowledge base, rendering its future behavior more indeterministic over time (Hancock, 2019). The relative functional stasis of automated systems compared to the dynamism of autonomous systems is what differentiates the two. However, much like how animals are born with an innate set of reflexes and the ability to learn and develop their independence, so too may autonomy emerge in an automated system from an initial set of knowledge and action capabilities (Hancock, 2019; Russell & Norvig, 2014). Despite the definitions provided above, "automation" and "autonomous systems" are often conflated (Hancock, 2019; Kaber, 2018). The reason for this is because *automation*—besides being defined as a process—can also refer to *technology*, i.e., the mechanized or computerized systems designed during the automation process (Kaber, 2018; Parasuraman et al., 2000). This terminological mix-up has resulted in decades of conceptual misunderstandings and discussions (some of which will be outlined in this paper). On a positive note, it has also generated several different theoretical and applied frameworks for Human-Automation Interaction (HAI) research and design.

Several observations can be made following the previous discussion. First, most HAI literature refers to automation as a technology (artifact) rather than a process (e.g., Billings, 1996; Endsley, 1995; Parasuraman et al., 2000; Wickens et al., 2013; Woods, 1985). Second, most HAI literature uses the terms automated/automatic and autonomous interchangeably (e.g., Norman, 1993; Wickens et al., 2013). Third, the qualities of automation technology as discussed in HAI literature are typically more aligned with the definition of an automated/automatic

system (as defined above) than an autonomous system (i.e., not all automation systems are autonomous systems). Fourth, given that automation systems can have different properties and integration effects (see Figure 2 for an example) it is important for product/system designers and Human Factors engineers to consider: what kind of activity the automation system will be a part of; who will interact with it; in what context; for what purpose; and in what ways. For various contextual reasons (technical, financial, regulatory, etc.) a proposed automation system in a given domain (e.g., air traffic control, self-driving vehicles, or even a burglar alarm) may require automated or autonomous functionality. In this way, automation is not a unitary concept - there are many ways to implement automation. Therefore, the nature of automation technology is meaningless unless considered in the context of a specific task. This needs to be an informed design choice to ensure that the system is safe and efficient.

For the purposes of this report, we define automation as a dual concept involving both artifacts and processes. Firstly, automation refers to a range of technological, mechanical, digital, or cognitive artifacts designed to either partially or fully replace human involvement in specific functions, tasks, or processes, while simultaneously enhancing task performance. These artifacts encompass various technologies, including robotics, software automation, machine learning models, and cognitive assistants. Secondly, automation encompasses the process and activity of designing and integrating these artifacts into existing workflows or systems. This process aims to optimize various activities by streamlining operations, improving efficiency, and enhancing overall performance. Whether introducing robotics to manufacturing or integrating machine learning algorithms into decision-making processes, automation involves strategic planning and implementation to achieve specific outcomes.

	Agent type	
Characteristic	Automated	Autonomous
Role (in context)	• Partial mapping of human activity in context • Demand human coordination for task success • Requires human formatting of inputs and adaptation to outputs	• Maps human role in mission • Capable of adapting to various forms human input and adapting output for human use
Viability/reliability	• Lacks 'perfect' reliability • Function limited to specific operating conditions • Requires human monitoring and intervention	• Viable for identified task environment • Reliable for defined 'window of operation'
Impact on human behaviour	• Human supports agent in function (with hope of greater performance)	• Agent supports human partner in achieving broader mission
Impact on human workload	• Imposes alternative forms of workload (cognitive vs. visual motor) for human supervisor	• Agent imposes no additional workload on human partner in command/coordination of activities
Collaboration with human	• Requires supervision in assisting human (including function initiation, performance monitoring, intervention in error states)	• Operates in partnership with human to achieve common goal (including information sharing and division of task responsibility)

Figure 2: The contextual roles, viability, and impact on human behavior, workload, and collaboration of automated versus autonomous agents (Kaber, 2018).

2.2 Automation Incentives

A strong driver behind automation in safety-critical systems has been to address the issue of 'human error' - the perception that the decisions and actions of people contribute strongly to failure in industrial contexts. Indeed, 60 to 90 percent of major accidents and incidents in high-reliability organizations like nuclear power control or aviation are attributed to "human error" (Hollnagel & Woods, 2005; Wickens et al., 2013). This has reinforced the notion of the human operator as being an unreliable agent whose role should be reduced to monitoring otherwise autonomous or automated systems. From this perspective, designers are system custodians tasked with protecting safe systems from the unreliable and erratic behavior of humans who

get tired, stressed, irritable, or distracted due to cognitive limitations in terms of perception, attention, memory, information processing, decision making, etc. The central idea, then, has been to substitute the human operator with automation systems that requires minimal human involvement, thus reducing or eliminating catastrophic human errors. However, although humans make mistakes, err in judgement, forget, and slip up during task performance, they often do so due to poor system design and inadequate organizational support functions rather than carelessness (Gao & Dekker, 2016; Wickens et al., 2013). In fact, there are many notable instances of decisive human actions — rehearsed or improvised on the spot (Meshkati & Khashe, 2015; Trotter et al., 2013) — detecting, mitigating, or averting major accidents or disasters. Examples include the 2009 landing of US Airlines flight 1549 on the Hudson River (Meshkati & Khashe, 2015), the 1983 glide landing of Air Canada Flight 143, known as the "Gimli Glider" (Reason, 2008), or the successful return of Apollo 13 in 1970 (Reason, 2008). For more routine performances, some have called into question the usefulness of the term "human error", as the same human action can lead to task success or an accident - the context is what changed (Dekker, 2002). Perhaps, then, a more fruitful alternative to "human error" is to consider human *performance variability* (Hollnagel, 2002; Hollnagel, 2013). Nevertheless, automation efforts have been largely successful in, for instance, the aviation domain where the push from industry and government entities to increasingly automate flight systems and tasks (e.g., autopilots, flight directors, calculating fuel-efficient routes, navigation, or system malfunction alerts) have benefitted pilot workloads and safety margins (Billings, 1996; Wiener, 1988). Similarly, the ground proximity warning system (GPWS) has dramatically reduced terrain strike accidents since its introduction by congressional mandate in 1974. It is clear that automation technology can greatly improve safety in aviation and other domains.

The interest in automation has many other explanations. For instance, automation can perform tasks that are too difficult or dangerous for human operators to perform, such as complex mathematical calculations, aerospace booster rocket stability control, or cleaning up hazardous materials (Wickens et al., 2013). Other automation systems perform tasks that humans are not cognitively equipped to do very well; tasks that require more vigilance, endurance, or workload capacity than operators possess. Examples include autopilot functions or various monitoring alert systems. In addition, automation can be used to augment (rather than replace) human performance physically or cognitively in such tasks (Billings, 1996; Wickens et al., 2013). For instance, a pilot flying without mechanical, hydraulic, electrical, or digital aids must fly at low altitude to avoid hypoxia, with considerable muscle power yet precise inputs over long periods of time to operate control surfaces, stabilize the aircraft, and adjust for changing aerodynamic conditions while maintaining visual ground contact to be able to navigate precisely. Automation technology allows designers and pilots to overcome such limitations and fly higher, faster, and in any weather. Similarly, human perception, situation awareness (SA), and decision-making capabilities can be enhanced through data collection, processing, and visualization tools. This approach to improving human performance can be observed in e.g., aviation, power plants (process control), or the medical field (diagnosis, patient status). Increased productivity and throughput are other common arguments for introducing automation in air traffic control, various medical applications, manufacturing, or Unmanned Aerial Vehicle operation (Wickens et al., 2013). Automation is also used to address economical concerns. In aviation, automation has greatly contributed to reduce flight times and fuel costs by introducing more efficient route plans or climb and descent patterns (Curry, 1979; Feazel, 1980). As in many other industries, labor costs constitute a large component of airline operating costs. Although automation has already been used to reduce the required number of cockpit crew members (e.g., navigators or flight engineers are deprecated roles), it is not clear what additional personnel reductions automation can enable. Flight time optimizations have also contributed to reduced direct labor costs. Efficient use of maintenance equipment, through automation, can also save costs in aviation. Although automation

technology is expensive, through acquirement/development, training, and maintenance, it appears to be a worthwhile investment.

Many factors seem to be driving this growing automation of systems. Today, it's clear that automation can improve efficiency and safety, and reduce costs in many areas. However, such benefits should not hide the transformations generated by this automation.

2.3 Automation in sociotechnical systems

Much of today's safety research is founded on a "sociotechnical" understanding of work. This focuses on the interaction between people and technology and the way that these interactions are embedded in social settings, including the way that work is managed, planned, organized, performed, and regulated (Hollnagel & Woods, 2005). The discipline of Human Factors works to understand how these interactions can be described and improved, often through the use of human-centered design methods (McCafferty et al., 2004). For these purposes, attributions of error have little value on their own but should instead serve as a starting point when looking for ways of improving work system performance (Hollnagel, 1991). In the words of Erik Hollnagel (1991), this can be seen as a process of "amplifying human strengths rather than reducing human weaknesses" (p.6).

In work processes where people and technology engage in close interaction, technological changes can have both positive and negative consequences, and some of those consequences may have implications for safety. Work roles may appear, disappear, or transform, standard routines may no longer apply, and new paths to failure may emerge (Woods & Dekker, 2000). In a longer perspective, changes like these may set off a series of required calibrations. When the context of work changes, so will people's activities. In turn, changes in working patterns may demand new technological adaptations, thus continuing a cycle of adjustment (Carroll et al., 1991). These ideas around human-machine interdependence have had a profound impact on the issue of automation. As previously discussed, a common ambition in automation projects is to replace human activities partly or fully with automated functions. If in those cases, human contributions to safety, efficiency and effectiveness are poorly understood, they also run the risk of being underestimated. Because changes to technology are also prone to affect other parts of the work system, e.g., how people work and interact, simple substitution may prove to be an impossibility. This scenario—and the potential pitfalls it is associated with—has been described in the literature as the "substitution myth" (Sarter et al., 1997).

Researchers within the safety field were early to point out potential hazards in automation approaches that build on the idea of substitution. As automation in a work system increases, people in the process may be driven into a more passive role, a situation where both vigilance and competence may suffer (Endsley, 2017). Firstly, a more passive role may decrease the readiness of human operators to respond to anomalies. Secondly, solving problems in a highly automated environment may require skills that are difficult to uphold as the role of the operator transforms. The pattern where expected safety benefits of automation instead turn into system vulnerabilities has been described as an 'irony of automation" (Bainbridge, 1983), often discussed in terms of human "out-of-the-loop" (OOTL) performance problems (Endsley & Kiris, 1995; Kaber & Endsley, 1997). Automation is frequently accompanied by a decrease in operator performance, such as a reduced sensitivity to important signals (Billings, 1991; Wiener, 1988), excessive or insufficient trust in system ability (Parasuraman et al., 1993), and loss of operator situation awareness (Carmody & Gluckman, 1993; Endsley, 1996; Endsley & Kiris, 1995). A major consequence of the OOTL performance problem is that operators of automated systems may be unable to take over manual operations in the case of automation failure. Particularly, the OOTL performance problem causes a set of difficulties including a longer latency to

determine what has failed, to decide if an intervention is necessary and to find the adequate course of action (Billings, 1991). Findings of the same nature have also been made within the aviation domain. For example, Berberian et al. (2012) explored how the pilot's sense of agency (i.e., sense of control over their own actions) was affected by different degrees of autopilot assistance in a flight simulator. The results showed that the sense of agency decreased with the level of automation involved. The authors argued that the increasing level of automation tends to distract operators from action outcomes, decrease their sense of control and therefore disrupt their overall performance. In addition to direct effects on human-technology interaction, the benefits of automation solutions may also be limited by social or emotional responses from humans. One common issue is the attitude of trust in automation, i.e., the perceptions that people have around the capabilities of automated systems. If a system is over-estimated, human operators may run the risk of missing situations where their intervention is necessary. Conversely, if automation is under-estimated it may limit its use, restricting its benefits (Lee & See, 2004).

Nowadays, automation is a critical component of safe, reliable, and efficient industrial operations. When properly introduced, automation has the potential to offer both better working conditions and increased productivity. For decades, automation has been driven by technological considerations. However, a "work-driven" design approach is increasingly being advocated (e.g., Miller & Feigh, 2019), as many studies have highlighted the limits of technology-driven automation. These limits are particularly related to the involvement of human operators in the control loop, as well as to their understanding of the current situation (see for example Endsley, 1996; Endsley & Kiris, 1995). It is therefore necessary to find effective principles and methods for distributing functions between the human and artificial system components to ensure that both operate in an optimal way. In the following sections, we introduce some of these solutions.

2.4 Function Allocation

From the description of autonomous and automated agents in section 2.1, automation can be classified on a continuum of levels of human replacement, ranging from manual operations to the highest level of so-called "full" automation where the human is ignored. Several different scales have been suggested in the literature to represent this continuum of levels in what is commonly referred to as "Levels of Automation (LOA). One such example is given in Table 1, which is based on the LOA taxonomy by Sheridan and Verplank (1978). It consists of a ten-point scale where higher values correspond to reduction in human involvement. Other versions of LOA models can be found in the literature (see e.g., Endsley & Kaber, 1999).

Table 1: The ten levels of automation. Adapted from Sheridan (2011).

Level of Automation	Description
LOW	1. The computer offers no assistance: human must take all decisions and actions.
	2. The computer offers a complete set of decision/action alternatives, or
	3. narrows the selection down to a few, or
	4. suggests one alternative;
	5. executes that suggestion if the human approves, or
	6. allows the human a restricted time to veto before automatic execution, or
	7. executes automatically, then necessarily informs the human, and
	8. informs the human only if asked, or
	9. informs the human only if it, the computer, decides to.
HIGH	10. The computer decides everything and acts autonomously, ignoring the human.

An additional parameter of description, and a direct extension to the LOA paradigm is to combine levels of automation with *types* of automation (i.e., what kind of task does it perform, and how much it replaces the human operator in doing so?). One such approach is presented in Parasuraman et al. (2000) where the types are derived from a simplified four-stage model of human information processing:

1. **Information acquisition**: sensing and registering input data. Low LOAs can amount to mechanically moving sensors to scan and observe, e.g., fixed-pattern sky scan vs dynamic "lock-on" tracking. Moderate LOAs could involve organizing collected data according to criteria, such as electronic flight strips indicating problems with an aircraft through the use of highlighting (Parasuraman et al., 2000).

2. **Information analysis**: cognitive functions such as working memory and inference. At low LOAs, algorithms can make prognoses from new data based on historical records, e.g., predicted path trajectories. Higher LOAs involve the integration of multiple parameters into one, such as the converging runway display aid (CRDA) which saves the ATC operator from the taxing mental work of projecting the approach path of one aircraft onto the path of another aircraft landing on a converging runway (Parasuraman et al., 2000).

3. **Decision and action selection**: generating and selecting among decision alternatives. This could involve simple conditional rules to select specific decisions if certain conditions apply, such as in route planning for pilots to avoid bad weather. Inferences

based on implicit or explicit assumptions are made to compare decision costs and prospective value. Compared to ground proximity warning systems (GPWS), which recommends a single optional maneuver, an automatic ground collision avoidance (auto GCAS) system takes control of the aircraft if the pilot does not respond (Parasuraman et al., 2000).

4. **Action implementation**: executing a selected action. These automation technologies often replace the human hand or voice. At increasing levels of automation, a photocopying procedure can include manual sorting, automatic sorting, automatic collation, automatic stapling, etc. In a more complex setting, ATC "handoff" of an aircraft from one flight sector to another can be done automatically once decided upon by the ATC controller. Similarly, on the flight deck, updated flight plans can be automatically loaded into the plane's flight management computer once uploaded from the ground, as opposed to entered manually (Parasuraman et al., 2000).

A function that is classified to any of these types can then be automated to different degrees or levels.

1

Deciding which functions (tasks, jobs, responsibilities) in a man-machine system should be assigned to the operator or to the machine is a central element of systems engineering (Hancock & Scallen, 1996; Price, 1985). All the more since it has been shown that an intermediate level of autonomy can compensate for the phenomenon of loss of autonomy (Endsley & Kiris, 1995; Kaber & Endsley, 1997).

This process is known as *function allocation*. It has previously been defined as follows:

> Allocation of function is an early stage of the design of a human-machine system. The input to allocation of function is a specification of the functions that the human-machine system must deliver within its intended working context. The output from allocation of function is a specification, at an appropriate level of abstraction, of the functionality of the automated subsystems that will be required. The goal of allocation of function is to design a system for which: the performance (including considerations of safety and reliability) is high; the tasks of the operator are achievable and appropriate to the operator's role; and the development of the system is technically and economically feasible. (Dearden et al., 2000, p.289-290).

Thus, function allocation techniques aim to provide strategies for distributing system functions and tasks across people and technology.

In 1951, Fitts's list (Fitts, 1951) marked the beginning of function allocation research, and seven decades later it continues to be one of the primary methods used in human factors. Sometimes criticized, it remains one of the first steps in the design of many systems. Understanding this method—its advantages and limitations—appears critical when it comes to proposing a methodology for designing autonomous human systems.

2.4.1 The HABA-MABA Methods

If several methods are currently proposed, most of them are based on the compensation principle initially proposed by Fitts (1951) where the strengths and weaknesses of humans and machines serve as the basis for assigning functions and responsibilities to the various components of the system. Specifically, this classic function allocation method consisted in sharing the tasks to be carried out between man and machine according to the strengths and weaknesses of each. Originally, Fitts compiled a list of eleven statements about whether a

human or a machine performs a certain function better (see Figure 3). In its literal interpretation, Fitts' list recommends that those functions that are better performed by machines should be automated, while the other functions should be assigned to the human operator. Fitts' view was that, by applying these criteria, an optimal allocation of functions between humans and machines could be achieved.

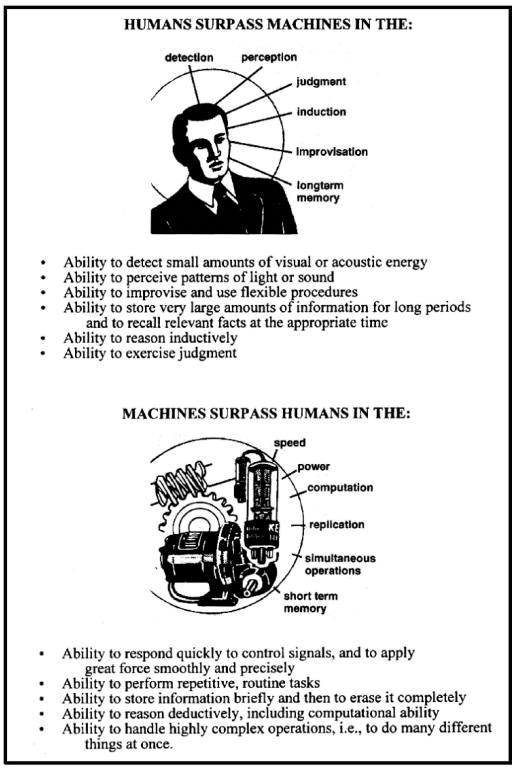


Figure 3: The Fitts list (Bradshaw et al., 2012; Fitts, 1951).

Typically, the input to these processes is a list of abstract functions that the human/machine system should perform, and the output is usually the same list ranked according to whether the human, the machine, or some combination should perform that function (Sheridan, 1997). The decision process considers, on the one hand, the functions that are technically possible to assign to machines and, on the other hand, the functions that humans can reasonably be expected to perform effectively. In this approach, human and machine are construed as actuating and information processing systems with different capabilities. This description can be used to determine what should be automated and what should not. This approach presumes that human and machine capabilities, strengths, and weaknesses remain fixed over time, and often suggests a quantitative division of labor where humans do a certain amount of work and machines do another (Dekker & Woods, 2002). The eleven statements initially proposed had several advantages. First, they were diverse and not contradictory. Second, they were based on solid theoretical foundations, notably concerning the limiting characteristics of human capacities and performances (overload, stress, fatigue, inattention, etc.).

Through this initial list, Fitts' goal was to propose a general framework to guide the allocation of functions, not to propose allocation principles that were intended to become dogma. According to this method, the critical point to focus on is performance: if the machine outperforms the human, the function must be automated; otherwise, automation is meaningless. Thus, Fitts' list indicates that the main (but not necessarily the only) driving force for automation must be performance: accuracy, power, speed, cost. In this sense, it is obvious that this list was bound to evolve as technological progress modulated the performance of the

systems. Thus, HABA-MABA lists, or 'Human Are Better At - Machines Are Better At' lists have appeared over the decades in various guises (e.g., Chapanis, 1965; Mertes & Jenney, 1974; Swain & Guttman, 1983; Sheridan, 1997).

Today, numerous fine-grained function allocation models can be found in the literature. The HABA-MABA approach benefits from the comprehensibility and simplicity of Fitts' list. It does not contain complex equations, interconnected functions, or other forms of complexity. Such a method also has been proved to increase both performance and situation awareness. Furthermore, it applies to a range of different functions, both physical and mental (De Winter & Dodou, 2014). Moreover, it encourages designers to consider the strengths and limitations of both humans and technology elements in a system. To summarize, the simplistic nature of Fitts' list provides a good starting point for any process of design. However, several limitations should be noted.

First, HABA-MABA methods looks at the strengths and weaknesses of each actor in a very static way (see Dekker & Woods, 2002; Bradshaw et al., 2012) and have few considerations about the fact that different functions can have different allocations over time. This view was summarized in a report from the US Department of Defense:

> An ... unproductive course has been the numerous attempts to transform conceptualizations of autonomy made in the 1970s into developmental roadmaps. ... Sheridan's taxonomy [of levels of automation] ... is often incorrectly interpreted as implying that autonomy is simply a delegation of a complete task to a computer, that a vehicle operates at a single level of autonomy and that these levels are discrete and represent scaffolds of increasing difficulty. Though attractive, the conceptualization of levels of autonomy as a scientific grounding for a developmental roadmap has been unproductive. ... The levels served as a tool to capture what was occurring in a system to make it autonomous; these linguistic descriptions are not suitable to describe specific milestones of an autonomous system. ... Research shows that a mission consists of dynamically changing functions, many of which can be executing concurrently as well as sequentially. Each of these functions can have a different allocation scheme to the human or computer at a given time. (Defense Science Board Washington DC, 2012, p.23-24)

Rouse also pointed out this limitation:

> Frustration with the MABA-MABA approach led to a very simple insight. Why should function, tasks, etc. be strictly allocated to only one performer? Aren't there many situations whether either human or computer could perform a task acceptably? ... This insight led to identification of the distinction between static and dynamic allocation of functions and tasks. (Rouse, 1994, p. 29, as quoted by Inagaki, 2003)

More recently, dynamic allocation methods have been proposed (e.g., Byrne & Parasuraman 1996; Greenstein & Lam 1985; Hancock & Scallen, 1996; Scerbo, 2007). The principle here is that not all allocations are applicable to every situation encountered by the system. Therefore, it is beneficial to alter the allocation according to certain situational factors. This is particularly applicable when a task can be performed equally well by human or machine and the allocation can therefore vary according to the available resources of each one (Rouse, 1981, Greenstein & Lam, 1985). Today, different forms of dynamic task allocation exist, such as "situation dependent" (Greenstein & Revesman, 1986), "flexible" and "adaptive" allocation (Mouloua et al., 1993), and "adaptive aiding" (Rouse, 1988).

A second criticism of the HABA-MABA approach concerns the fact that automation changes the nature of human activity. System automation has been classically considered as a simple substitution of a machine activity for human activity, referred to as the "substitution myth"

(Sarter et al., 1997). Unfortunately, such assumption corresponds to a distorted reflection of the real impact of automation. Automation technology transforms human work and forces people to adapt their skills and routines (Dekker & Woods, 2002). Whatever the merits of any particular automation technology, automation does not merely supplant human activity but also transforms the nature of human work. In future systems, automatic devices will provide for the real-time, moment-to-moment control of the process. In such systems, the main role for humans is to undertake what is called supervisory control (Moray, 1986; Sheridan & Verplank, 1978). Indeed, the human operator is relegated to the role of monitor and decision maker, keeping watch for deviations and failures, and taking over when necessary. This is a new relation between the human and the machine, as an automatic machine may be said to be intelligent. The new form of interaction differs dramatically from the traditional interaction of the human with tools and devices that possess no intelligence, in which all sensing and control were done by the human operator. Such change (from manual to supervisory control) is far from trivial. The role of passive information processor (i.e., that of supervisory controller) involves observing the actions of other operators or computer controllers and agreeing or disagreeing with them. The operator's task is to understand the actions of another system controller and thereby accept or reject its actions. The key difference between passive information processing and direct action on the process is that the former involves functions similar to those maintained during process monitoring (e.g., scanning information sources); whereas the latter involves manual control functions including process planning, decision making, selecting responses, and implementing strategies. The problems due to automation are related to these new roles that are created for operators when their tasks are changed from manual to supervisory control.

A third point of criticism is that HABA-MABA approaches do not account for man-machine cooperation. Creating partially autonomous machine agents is, in part, like adding a new team member. One result is the introduction of new coordination demands and the emergence of new classes of problems which are due to failures in the human-machine relationship. Many challenges facing human-machine interaction designers involve teamwork rather than the separation of duties between the human and the machine (Klein et al., 2004). Effective teamwork involves more than effective task distribution; it looks for ways to support and enhance each member's performance. This need is not typically satisfied by HABA-MABA-based function allocation methods.

Despite these—and other—criticisms, Fitts' list (or variants thereof) has been a widely used function allocation technique (Older et al., 1997), but other function allocation methods have been proposed to overcome its limitations.

2.4.2 Other function allocation methods

Parasuraman et al. (2000) provide a process template for designing automation systems (see Figure 4). In this framework, a (re)design process begins by deciding what to automate. The target task is categorized according to the four automation types and a target LOA is selected. This initial type and LOA configuration is evaluated against primary criteria—including human performance metrics like mental workload, situation awareness, etc.—and secondary criteria like automation reliability, cost, and so on. Designs that fail with respect to these criteria trigger a redesign, and the processes starts over.

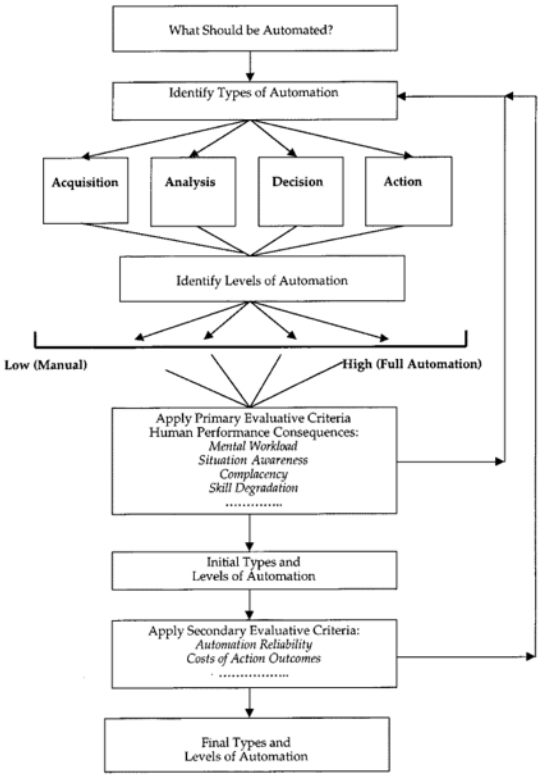


Figure 4. Flow chart of automation design using types and levels of automation (Parasuraman et al., 2000).

The appeal of this framework lies in its readability and streamlined application, but it also has its drawbacks. First, it assumes that the initial idea—i.e., the task or function to be automated—is fully understood and *can* be automated without negative impact on human performance, safety, and overall system performance. Rather, these and similar insights are developed in evaluation procedures after the automation design phase, at which point considerable time and resources may have already been spent on designing the automation in the first place. A second problem with the LOA framework is that it implies that the LOAs are *prescriptive*, i.e., that a system designer should target a specific LOA and design to "achieve" it, thus running the risk of neglecting proper user research and developing inadequate designs. In our view, LOAs are *descriptive*: they can be used to convey the capability of a system or design in an easily accessible way. Finally, the original LOA framework (and its derivatives) can be improperly applied to categorize a system holistically which says little about the capabilities of the system on a task-by-task basis, much like how a hotel receiving an average review score of 4.3 out of five says nothing about the quality of the menu or the cleanliness of the rooms, specifically. The extension framework by Parasuraman et al. (2000) mitigates this third problem by breaking the system down into automation types.

2.5 Techniques and methods to improve human-automation interaction

If allocation methods introduced in the previous chapter were proven to be helpful in the design of new automation systems, they lacked in the consideration of the interaction and coordination between the agents of a human-machine systems leading to serious limitations.

Rather than focusing on automation technology (or LOA) or human operators, this section will consider approaches developed to model, understand, and design the interactions between automation systems and human-automation.

2.5.1 Task Analysis

Since its inception in 1967, Hierarchical Task Analysis (HTA) has become an important tool in the human factors and cognitive engineering disciplines (Salmon et al., 2010; Stanton, 2006). In contrast to behavioral or psychometric constructs used in other task analysis paradigms, HTA focuses on *functional* system constructs. Furthermore, as the name suggests, HTA is about task analysis, not merely task description. As such, HTA is about identifying problems in system performance and proposing solutions. Overall system performance (i.e., of the joint cognitive system; Woods, 1985) can effectively be improved by addressing the factors responsible for the largest error variance, be it humans, machines, or effects emerging from their interactions (Annett, 2004).

The HTA process (illustrated in Figure 5 begins by defining functional task *goals*. Goals provide a purposeful frame to ground the tasks, as goals can often be attained in multiple different ways, particularly in complex systems. A specified goal state—active or latent—can be an event, value, or other criteria that constitute goal attainment and system performance (Annett, 2004). In other words, the system's objective is implied by the goal(s) (Stanton, 2006). Through the process of *decomposition* or *redescription*, complex goals are broken down into subgoals which enables the analysis of actual or potential sources of system failure in goal attainment (Annett, 2004).

Operations are the fundamental units of analysis in HTA and are defined through their goals (Annett, 2004). Operations—like goals—can be further broken down into suboperations (or subtasks), organized in a nested hierarchical structure (Annett, 2004). Suboperations make unique contributions to superordinate operations and subgoals, ultimately contributing to the attainment of superordinate system goals (Annett, 2004). Suboperations are mutually exclusive and should provide a comprehensive analysis of sub- and superordinate goals (Annett, 2004).

Plans are a critical component of HTA. Where goals state the purpose of tasks, operations describe the content of tasks, plans describe the order of task and operation performance. There are different types of plans: simple task sequences ("do this, then this"): selection rules ("if x is the case, do this. Otherwise, do this"); or dual task or parallel plans ("do this and this at the same time") (Annett, 2004). An example HTA diagram is presented in Figure 6.

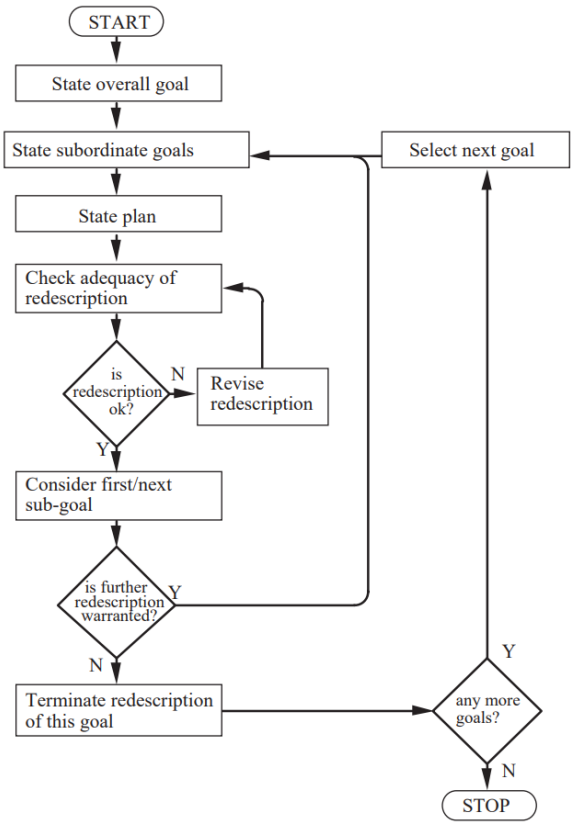


Figure 5. HTA procedure for breaking down the sub-goal hierarchy (source: Stanton, 2006).

A challenge when producing an HTA is knowing when to stop, as the analysis could continue almost in perpetuity. A common heuristic is to stop when it is assessed that further redescription will not yield additional insights for the purposes of the analysis objective. Another slightly more formal stop rule is the "*P* x *C*" rule, which states that an HTA is complete when the product of the probability (*P*) and cost (*C*) of failure is within acceptable bounds (Annett, 2004; Stanton, 2006). In theory, this rule helps the analyst to focus their attention on the task factors that are key to overall system performance and success. In practice, *p* and *c* can rarely be known, only estimated, but their product is what informs a decision to stop or continue the analysis. In error or accident investigations, the analysis naturally concludes when the analyst can provide a satisfactory explanation about the cause, and propose design remedies in terms of systems, procedures, or training interventions (Annett, 2004).

Stanton (2006) highlights two important points about HTA. First, it is a goal-based analysis, as discussed above. Second, HTA produces a *system* analysis. This means that the analysis is not exclusive to human agents but can also model tasks performed by non-human agents (e.g., equipment, devices, and interfaces) and teams (Salmon et al., 2010; Stanton, 2006).

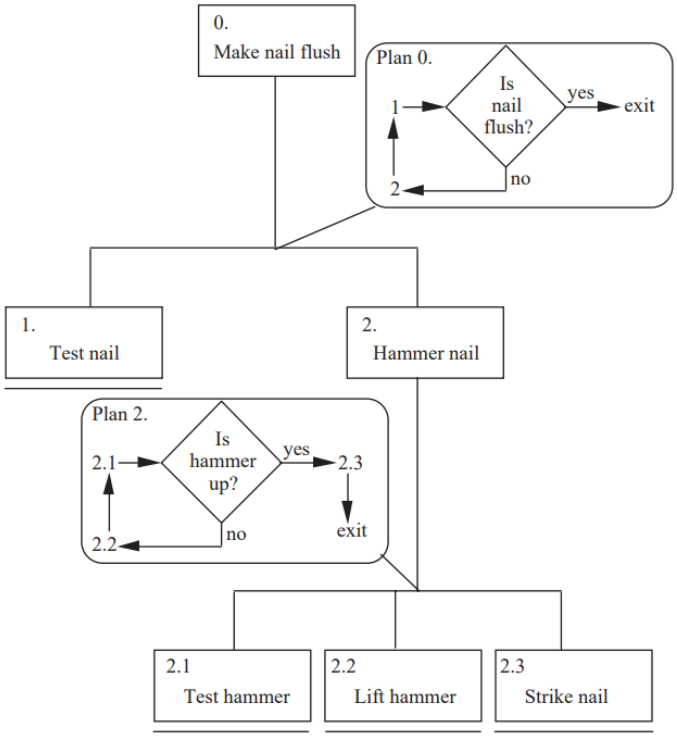


Figure 6. Example HTA for making a nail flush with a board (source: Stanton, 2006).

HTA is a simple (but not easy) process where data is collected through interviews, observations, questionnaires, walkthroughs, user trials, etc., about the system or task in question, and then used to decompose the tasks into goals, operations, plans, and conditions. This simplicity contributes to the method's flexibility, which has seen HTA being adapted, extended, and applied in many ways. Application areas include examples such as interface design, human error prediction and analysis, team task analysis, function allocation, workload assessment, procedure design, and the design and development of nuclear reactor plants (Salmon et al., 2010; Stanton, 2006).

2.5.2 Coactive Design

When an automation system reaches a certain degree of *self-sufficiency* (i.e., capability in the task to be performed) and *self-directedness* (i.e., authority over the task to be performed), several issues may arise. First, in a fully teleoperated system these are entirely absent, rendering it a burden to the operator. Second, a system with low self-sufficiency but high self-directedness will be over-trusted, increasing the risk of system failure. Third, and conversely, a system with high self-sufficiency and low self-directedness will be under-utilized. This is common in cases where the cost of failure is deemed too high. Finally, highly self-sufficient, and self-directed systems tends to become opaque in their system states and action performance, making it difficult for team members to maintain a sufficient situation awareness and adjust their own task performance accordingly (Johnson et al., 2011b).

To remedy this, Coactive Design takes a **teamwork-centered** approach to systems design and task allocation. It introduces *interdependence* as a third autonomy dimension. Its premise is that processes of understanding, problem solving, and task execution are necessarily incremental, subject to negotiation, tentative, and of critical importance to joint activity between people and autonomous systems (Johnson et al., 2011b).

Interdependence, as a central organizing principle for human-agent joint activity, characterizes how actors depend on the actions of one another over a sustained action sequence (Johnson et al., 2011b). Dependence, in a strict sense, could be a locomotive engine pulling a train cart. In a softer sense, optional, opportunistic, helpful, and *mutual* support actions are emphasized (i.e., *inter*dependence). Soft interdependence is frequently observed in human teams, like partners offering to pick up items at the store on their way home from work, colleagues informing other that they will run late, or opening a door for a stranger who has their hands full (Johnson et al., 2011b). In addition, to facilitate supportive (i.e., intra-activity) interdependence, sufficient monitoring capabilities are required. People and artificial agents must monitor the contextual aspects of tasks performed by others to "look out for each other" and provide timely assistance. This, in turn, requires the monitored actor to be transparent in their behavior (Johnson et al., 2011b).

Joint activity and interdependence require a minimal level of autonomy (i.e., self-sufficiency and self-directedness) of its participant actors, yet increased autonomy, depending on its implementation, can boon or cripple team performance (Johnson et al., 2011b). Rather than designing for a human-out-of-the-loop paradigm, coactive design enables close and continuous human-automation interaction. It contends that to effectively engage in joint activity, an agent must be aware and considerate of the interdependencies of their joint activity and have the capability to support it (Johnson et al., 2011b). The **awareness** requirement highlights that joint activity, as a mutual engagement, extends in space and time. Where agents were previously only concerned with their own allocated tasks, the increasing sophistication of joint human-computer systems will require agents to have a matured understanding of their role and actions, their interdependencies, and the goal of the joint effort. In this sense, it is about group *participatory* actions rather than individual *autonomous* actions (Johnson et al., 2011b). The **consideration** requirement describes how an agent's autonomous capabilities must be designed with respect to the needs of the team or collective. The same action outcome can result from different processes and performance dynamics, like performing a musical solo piece versus a duet. Artificial systems and their behavior must also reflect this. Additionally, joint activity, compared to individual action, introduces additional constraints. Collective obligations must be considered and tended to, even when not assigned to any particular agent. In this way, participants share an obligation to coordinate which may also require them to sacrifice their own autonomy in service of group goals. These obligations come at a cost *and* provide benefits (Johnson et al., 2011b). The requirement for coactive agents to have the **capability to support** interdependence describes how agents must have the ability to provide and receive assistance. Reciprocity is therefore a functional requirement of coactive agents to facilitate good teamwork. If one agent needs to know the status of another, the second agent must be able to provide it, and vice versa (Johnson et al., 2011b). In addition to its self-sufficiency and self-directedness, the capability of a system or agent to support interdependence offers multiple benefits. For instance, over-trusted systems can aid, and otherwise opaque systems can provide appropriate transparency and feedback to human operators (Johnson et al., 2011b).

Coactive design challenges the prevalent notion that more autonomy equals more performance, which is not accurate as previously introduced. Instead, the authors contend—and empirically support—that "there is a point in problem complexity at which the benefits of autonomy may be outweighed by the increase in system opacity when interdependence issues are not adequately addressed" (Johnson et al., 2011b, p.186). The challenging future roles in human-agent systems will require more interdependence than what is typically considered.

This necessitates a design shift away from who is in control of whom or who is tasked with what (i.e., function allocation), to *coactivity*. Coactive design provides prescriptive high-level guidance about the considerations required to design coactive human-agent systems (Johnson et al., 2011a, Johnson et al., 2011b). By understanding the dynamics and interdependencies of work, system designers and developers can make informed choices about what to automate, how to reduce operator workload and assess how work is affected by the introduction of new technology (Johnson et al., 2011a).

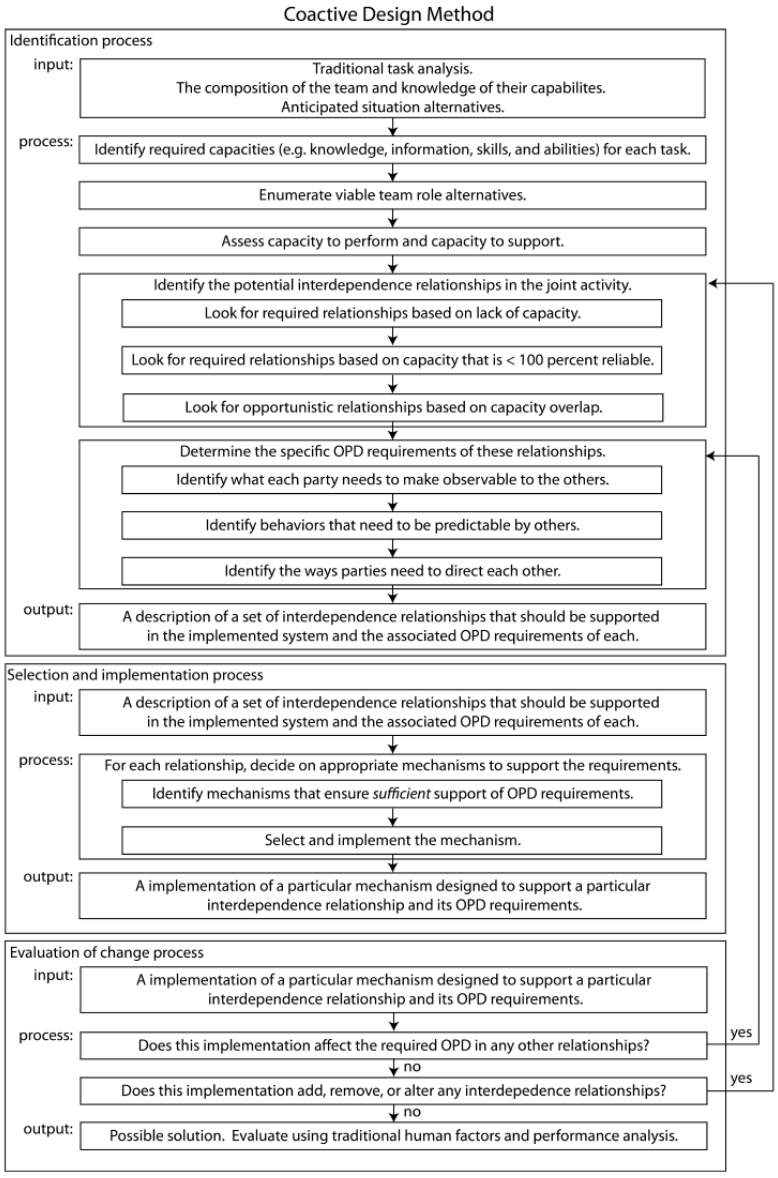


Figure 7. The Coactive Design workflow (Johnson et al., 2014).

This iterative method (summarized in Figure 7) begins with an **identification process** where designers extend a traditional hierarchical task analyses (HTA) to describe required capacities (cognitive task analysis; CTA) and viable team member roles (Interdependence Analysis; IA). The enumerated role alternatives are evaluated, and interdependence relationships are identified. Observability, predictability, and directability (OPD) requirements are also developed to understand who needs to observe what from whom, who needs to predict what, and how team members must be able to direct each other (Johnson et al., 2014). Next, the **selection and implementation process** take the identified relationships and determines what

mechanisms meet the OPD requirements. Selection criteria can include sufficiency or leveraging synergy effects, for instance (Johnson et al., 2014). Finally, the **evaluation process** determines how the chosen mechanism affects the OPD requirements on other relationships. The mechanism can also affect system performance by adding, altering, or removing interdependence relationships. Therefore, each designed mechanism must be evaluated from a holistic systems perspective, providing feedback to the identification and/or selection and implementation processes in an iterative design and development loop. Traditional Human Factors and system performance evaluations can begin when a design solution has been approved from an interdependence standpoint (Johnson et al., 2014).

The Coactive Design method is appealing in many ways. It recognizes that humans and machines combine in a joint cognitive system whose purpose is to perform a function, and whose execution performance of that function does not rest on the performance of the human or machine is isolation but emerges from their interactions. By designing with these interactions and interdependencies in mind, several of the issues discussed in section 2.2 are addressed. The method explicitly addresses the substitution myth by making the human contribution equally as important as that of the machine. Meaningful and stimulating tasks can potentially increase operator engagement and performance. The OOTL phenomenon—and its detrimental effect on performance—is mitigated as the human operator is highly involved in the (joint) activity, which can also potentially increase the operator's sense of agency compared to a situation of supervisory control. Skill retention is also ensured as the human may have to perform different steps of the task at hand.

The method also comes with a few notable drawbacks. For instance, it requires an initial and formal task analysis (H/CTA) to provide input to the identification phase. Such analyses often require much time and resources, thus incurring a considerable (or even prohibitive) economic cost. Another risk is introduced by the fact that the method does not account for the law of stretched systems (Woods & Dekker, 2000) which—combined with the fact that peoples' tasks, procedures, and use of technology evolve over time (Carroll et al., 1991)—suggests that the mechanisms designed to support the identified interdependencies and OPDs may become inadequate as users (and the system overall) adapts (to) the new technology. This can result in costly redesigns. Furthermore, if the operator, who is already engaged in critical teamwork activities, is additionally tasked with other responsibilities (as the law of stretched systems predicts they will) the risk of high temporal task demand, competing goals making operators' work challenging in new ways, challenges to sensemaking and situation understanding, and unwanted performance variability is increased which can jeopardize the team's performance. Another limitation is that the method does not recognize that conceptual limitations are encountered during—or can result from—technology development and implementation. In other words, there is no indicated feedback loop from the "select and implement the mechanism" step of the selection and implementation (S&I) process (or the S&I process overall) to the identification process; limitations of technology can necessitate a conceptual redesign.

2.5.3 Designing for Joint Action and a Sense of Agency

As previously explained, when implementing a high level of automation, a critical issue is the ability of the human operator to feel in control of those systems and/or of the actions performed in collaboration with those systems. Recently, some authors argued that providing access to different levels of intention implemented by AI could help restore human operators' sense of agency (SoA), improve their confidence in the decisions made by artificial agents, and ultimately increase acceptability towards such agents (Pagliari et al, 2022; Wen et al., 2022). These authors study joint actions in human social interactions to deduce what are the key features necessary to develop a reliable SoA in a social context. More particularly, they consider

the content of relevant explanations to be implemented in AI to make it "explainable". Their observation was that the sense of agency of co-authors of a joint action increases when those were sharing their intentions (Atmaca et al., 2008; Le Bars et al., 2020; Sebanz et al., 2003, 2005; Wegner et al., 2004; van der Wel, 2015; van der Wel et al., 2012). Based on this premise, it was explored how sharing the intention of the system supports the emergence of the operator sense of agency. As an illustration, Le Goff and colleagues (2018) explored how messages conveying the system's intent in supervisory situations. The idea was to display information about what the automated system is about to do next, which has been shown to be an effective approach to improve users' sense of control and acceptability towards the system. This study further demonstrated that providing information about a higher level of intention (P-intentions) than just the level of motor intention (M-intentions), increases the feeling of control of the action produced by an automated system, above and beyond improving bodily ownership. Such results emphasize the importance of the information provided by the artificial agents, especially to reduce their opacity. The importance of the communication, of the system's intentions, is also widely underlined when considering how to support cooperation between human operators and artificial agents, whatever their level of automation/autonomy.

2.5.4 Human-Autonomy Teaming Methods

The distinction between automation and autonomy, as addressed in this document, has led some authors to specifically question the problems of cooperation resulting from the autonomy of artificial agents. Recent technological evolutions have introduced a rupture in our interactions with technology. From simple tools at the service of the human operator, artificial agents have become full-fledged teammates characterized by a high level of autonomy in terms of decision making, adaptation and communication. Several researchers have explored to what extent and under what conditions autonomous agents and humans could work collaboratively in a team, leading to a new scientific field called Human-Autonomy Teaming (HAT).

Even if the term HAT emerged three decades ago, it is only during the last few years that it has been used frequently (e.g., Demir et al., 2018a; Demir et al., 2018b; Demir et al., 2019; Dubey et al., 2020; Fiore & Wiltshire, 2016; Grimm et al., 2018a, 2018b; Shannon et al., 2017; Wohleber et al., 2017). Interestingly, HAT papers (e.g., Chen et al., 2014; Grimm et al., 2018a; Demir et al., 2019) have also emphasized the role shared mental models and team situation awareness as cooperation enablers (or mediators, acknowledging the Input-Mediator-Output model from Kazi et al., 2019). In particular, these models have relied on the notion of transparency of artificial agents.

Transparency is defined as the descriptive quality of an interface pertaining to its abilities to afford an operator's comprehension about an intelligent agent's intent, performance, future plans, and reasoning process (Chen et al., 2014). Over the past decade, many studies have sought to demonstrate the role of this transparency. For example, higher levels of transparency can provide useful information for human decision making, thereby reducing the workload, or keeping it unchanged (Mercado et al., 2016; Selkowitz et al., 2016). Several studies have shown that as system transparency increased, human performance also increased, illustrating the benefits of transparency in terms of performance (e.g., Mercado et al., 2016; Stowers et al., 2016). Additional transparency could also improve situation awareness (Mercado et al., 2016, Selkowitz et al., 2016). Finally, transparency appears as one key element in establishing appropriate trust in the system (Sanders et al., 2014). In this context, different frameworks have been proposed to design more transparent systems, including the Situation Awareness-based

Agent Transparency model by Chen (2014, 2018) or the transparency on intentional, task, analytical, environment and teamwork agent's models by Lyons (2013).

The Situation Awareness-based Agent Transparency model proposed by Chen and collaborators (2014; 2018) is based on the concept of Situation Awareness (SA) proposed by Endsley (1995). Particularly, Chen and colleagues aimed to identify transparency requirements to enable the human operator to maintain proper SA of the system, i.e., to understand the parameters of the intelligent agent's task, its logic, and its expected outcomes. According to Endsley (1995), SA is "a person's state of knowledge of a dynamic environment" (p. 60). In Endsley's model, SA has three levels: perception (Level 1), comprehension (Level 2), and projection (Level 3). Accordingly, Chen and colleagues (2014, 2018) identified three levels of communication corresponding to the three levels of SA. At the first level of the SAT model, the operator is provided with the basic information about the agent's current state and goals, intentions, and proposed actions: "What's going on and what is the agent trying to achieve?". At the second level, the operator is provided information about the agent's reasoning process behind those actions and the constraints/affordances that the agent considers when planning those actions: "Why does the agent do it?". At the third level, the operator is provided with information regarding the agent's projection of the future state, such as predicted consequences, likelihood of success/failure, and any uncertainty associated with the projections: "What should the operator expect to happen?". By providing access to the underlying processes that the autonomous agent uses to make its decisions, actions, and projections, the agent allows the operator to build an efficient representation of the environment and its own functioning (Chen et al., 2018; Stubbs et al., 2007; Lee & See, 2004). This transparency of the agent has been identified as an essential factor in the development of appropriate SA in human-robot teams (Evans, 2012; Endsley, 2015). Also, Mercado and collaborators (2016) shows that increased transparency leads to increased performance on task "without additional costs" (on effectiveness and time).

In this search for system transparency, a complementary theoretical framework has been proposed by Lyons (2013) where transparency can be defined as a method to establish shared intent and shared awareness between a human and a robotic system. As for the SA-based Agent Transparency model proposed by Chen, Lyons also suggests the needed for transparency in Human-Robot Interaction to support human SA and developing "shared awareness" with robots. Among the variety of factors that determine human-robot team performance, communication between artificial agents and humans is according to Lyons one of the most important elements (see also Chen et al., 2006). Furthermore, this communication would have great similarities with the communication needed in human-robot teams (e.g., Salas et al., 2005).

Lyons (2013) highlights several models which should be transparent to humans: intentional model, task model, analytical model, environment model and teamwork model. The *intentional model* focuses on communicating to the operator the higher-level purpose of the technology, the method and style of interaction to be expected, the social/moral intentions of the technology, and some understanding of the technology's goal structure. This information should provide operators with some sense of general predictability about how interactions with the technology might occur while also giving them a sense of the system's priorities. The *task model* consists of the system's understanding of a task structure, information relating to the system's awareness of goals in relation to a task, information relating to the system's real-time progress in relation to those goals, and awareness of when the system makes a mistake. The *analytic model* provides operators with an understanding of how the system works, what calculations and algorithms it uses, and why it might make an error. The environment model should present operators with real-time information communicating the system's awareness of environmental conditions, constraints, and task related limitations in relation to the

environment. Finally, the *teamwork model* concerns information about the roles, responsibilities, and duties of one's teammates, about the team dynamics between the human operator and an autonomous system (Lyons, 2013). Lyons and collaborators (2017) suggest that these transparency facets will allow the human to understand the goals, social intent, contextual awareness, task limitations, analytical underpinnings, and team-based orientation of the system. This is an interesting precision of the notion of mental model presented earlier. Some of the suggested information, such as the information on the logic behind system's decisions—referring to the analytical model—has been proven to improve user trust (Lyons et al., 2017).

3 Method

This section will outline the methodology employed to develop the framework proposed in this deliverable.

3.1 Development criteria of the SafeTeam framework

The SafeTeam human factors framework is developed with the focus to be easily applied by practitioners. It is designed to assist readers to facilitate an optimized interaction between a human operator and an automated or autonomous system, with little to no prior knowledge of human factors.

Three criteria primarily informed and shaped the framework:

- The literature review of established theories and methods within human-autonomy interaction and function allocation.
- The study of SafeTeam use case 1: *En-route digital assistant* and application of the framework to the case.
- Continuous discussions and feedback sessions with SafeTeam use case 2: *Stabilized approach digital assistant*.

3.1.1 Literature review

The framework rely on evidence-based methodologies that has been merged and restructured to enable individuals without expertise to design and/or assess their automated systems with **human-machine collaboration** in mind. It is heavily influenced by task allocation model of Parasuraman and colleagues (2000) and Coactive design (Johnson et al., 2014), with a focus on task analysis through the creation and analysis of an HTA (Annett, 2004), and the consideration of observability, predictability and directability requirements in the design principles. The model is complemented with concrete instructions that guide practitioners through the different steps of the methodology.

3.1.2 Case study of use case 1

The use case *En-route digital assistant* informed the work through a semi-structured group interview and through the report D3.1 where an early iteration of the framework was used. The interview provided insights on work processes, how they consider HF and briefly what they experience as their challenges. The finished report illustrates both how the provided framework was used and its limitations, as well as providing the team with a detailed description of the use case for the team to work with. In addition, the team carried out own field studies at air traffic control centers in Sweden and France. Based on the deliverable and own observations, the team could apply the framework as it was developed to the *En-route digital assistant case.*

3.1.3 Collaboration with use case 2

For the second use case, *Stabilized approach digital assistant*, a more mature framework could be used in the writing process of report D3.2. In addition to the analysis of D3.2, the proposed methods were continuously discussed and improved as the team worked with the underlying

work presented in D3.2. The main intention of the discussions was to gain insight into work processes and attitudes towards human factors approach and to ensure usable methods and useful instructions. Results of D3.2 are used as examples in Chapter 4.

3.2 Creation of guidelines

Similar to the processes of coactive design (identification, selection and implementation, evaluation of change; Johnson et al., 2014), the guidelines are split in three phases, while encouraging iterations within and between phases.

A delimitation with the SafeTeam framework is that the guidelines do not cover any ideation methods that precede the first phase of the framework, a decision made based on the framework's alignment with SafeTeam's scope. Instead, the framework prioritize the establishment of a broad system understanding through phase I, *Modeling key system factors and interactions*, to which the initial idea (e.g., a design proposal, change requirement prompted by inefficient task processes, or inadequate artefacts) serves as input. When designing this phase, we initially identified system knowledge as a crucial step described by literature.

After providing a general guide for system understanding and analyzing the work performed in D3.1, more specific instructions were necessary to provide the support needed. Thus, we recommend the HTA as a tool key system factors and interactions, and to encourage the reader to map contextual factors related to their idea. The tasks included in the HTA will likely go beyond what might have been their intuitional and initial scope. To support the data collection during phase I, a separate general guide for conducting interviews was included in the framework. Interviews with stakeholders were deemed an important source of data while other methods for gathering data, such as focus groups, field studies, contextual inquiries, cognitive walkthroughs, were mentioned but not described in detail.

The second phase, *Designing for safe human-autonomy teaming*, revolves around the HTA just like the previous phase. When the researchers in the use-cases had gained an understanding of the context they were introducing changes into, we used the HTA as a tool to allocate new or changed tasks and investigate how these changes would affect the rest of the system. It was clear from following the work in D3.1 and D3.2 that support and guidance was needed in order to understand how to consider important human factors when doing the task allocation and detailing the idea. Thus, a set of discussion topics based on the literature provided in section 4.3.3.1, and insights from studying use-case 1 and 2 was introduced to phase II. Furthermore, seven design principles were introduced to assist the reader to continuously consider important human factors related to safe human-autonomy teaming while allocating tasks. The design principles are inspired by and draws from existing literature, as discussed in section 2. At the end of phase II, the reader is asked to transfer the HTA into a tabular format in order to facilitate documentation of the task allocation and its effects, and to provide a better format to keep working with the assessment of the proposed task allocation in phase III. The tabular format will also allow the designers of the new system to keep working with assessment and other design requirements outside of the human factors scope while still keeping track of the discoveries made during the three phases in the SafeTeam framework.

In phase III, *Assessing the design proposal for collaboration issues and risks*, the intention is to evaluate the proposed design and to identify risks related to human factors. To guide the reader to consider relevant human factors aspects when assessing their proposed design we

provide a list of questions, these questions are based on topics from the literature. The list of questions was developed after discovering the need in D3.1 and D3.2 had the opportunity to use list of risks as a starting point for assessing their design proposal, to narrow down the types of questions to ask we also label each task with a task type that maps to the question list. The risks identified are documented in the TTA along with proposals on how to avoid and mitigate the risks.

The reader is asked to iterate over different phases as they find necessary since it will be difficult for anyone to capture all aspects of task allocation and the effects at the first go. When the risk assessment in phase III is finalized, the SafeTeam framework leaves the reader with a table of task allocations, dependencies to other tasks, task types, potential risks, and mitigations for these.

4 The SafeTeam Framework

The proposed framework aims to facilitate a human-centric approach when designing automated systems and *Human-Autonomy Teaming* (HAT) settings. The SafeTeam framework is developed for non-experts and is based upon several other established guidelines, as detailed in previous chapters. Central to the proposed framework are the emphasis on the close cooperation among the involved stakeholders and the importance of early evaluation and feedback of human factors design principles throughout the entire development process. Frequent and continuous concept evaluations can reduce development costs and highlight design issues that can lead to automation and interaction issues. This framework is intended to improve human-autonomy teaming in high-level design work and does not contain guidance for detailed design work as, for instance, User Experience (UX) design.

4.1 Introduction

The framework is presented through three phases, which are centered around the tasks within a system. In addition to this, it is also important to keep in mind how additional system factors also influence the system and each other.

4.1.1 System factors

Because technology is not neutral, it will inevitably change any system into which it is introduced, as well as the intended (canonical) and unanticipated (exceptional) behavior of that system. People's understanding of what constitutes canonical and exceptional system behavior evolves over time. As a result, it is critical to understand the effects of change and how they might be foreseen and accounted for in design (Hollnagel & Woods, 2005). The features of technology include limitations, preconditions, and side effects that place additional change needs on the entities with which it interacts, such as people, technology, or society (Norman, 1993).

As every system encompasses many factors, the selection of the relevant ones will be guided by the consideration of six layers influencing the context at hand (Figure 8). At the center of the system, you will find the activities to be performed—the **tasks** (e.g., sending emails, writing reports). Those tasks are performed by **agents,** which can be human operators (e.g., store clerks, service technicians) or artificial agents (e.g., Virtual assistants, Chatbots). Those agents rely on **artifacts** to perform or offer support for their tasks. They can be physical tools (e.g., pens, hammers) or digital tools (e.g., databases, websites, software), rely on different modalities (i.e., visual, auditory, etc.), and display different properties (e.g., interactive, symbolic). These categories are not mutually exclusive, however; many artifacts may fall into multiple categories depending on their characteristics, functionality, domain, or context. The **processes** consider the interdependence between the tasks through their chronology and/or the interactions required between the agents (e.g., the production of an article written by a journalist, then reviewed by an editor, and later printed is constrained by this chain of tasks). These activities are conducted within and for other **organizations** (e.g., private companies, financial institutions, or educational institutions). They may differ in their hierarchical structures, departmentalization (e.g., finance, marketing, production), cultures (values, beliefs, norms), communication channels (e.g., emails, meetings, reports, social networks), decision-making processes (e.g., operational/strategic, degree of employee involvement, driven by data or intuition, etc.), and so on. Finally, organizational activities are governed by **regulations** (e.g., environmental regulations, data protection and privacy laws,

financial regulations, health and safety laws, intellectual property laws, consumer protection laws, etc.).

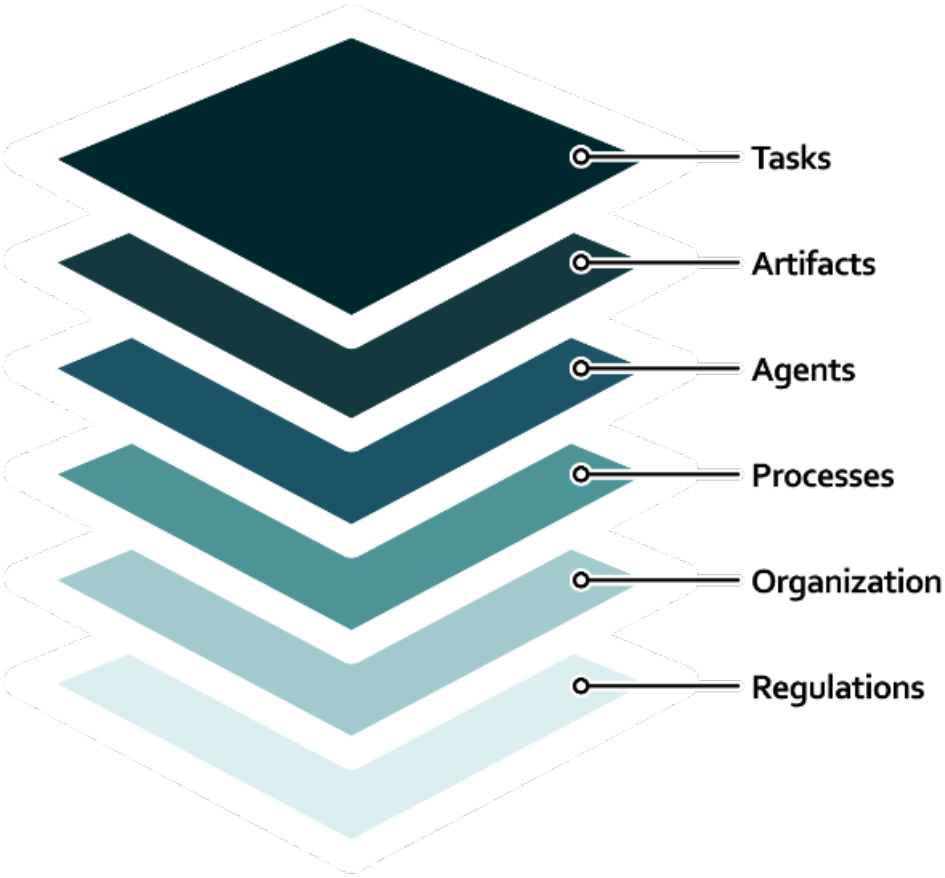


Figure 8. Factors of a system that can be affected and cause ripple effects.

The amount of detail may vary, but the focus of the approach is to distinguish elements that might be impacted by the implementation of a new function. More specifically, we want you to consider the **ripple effects** produced by the introduction of change in sociotechnical systems. Changes may impact the task at hand, the agent performing the task, the artifacts required to perform the task, the processes between agents of the system, the organization where the task is taking place, and even the regulations considered for this system. For instance, replacing a human customer service agent with a chatbot will likely generate new human tasks elsewhere in the system (e.g., a software engineer optimizing the performance). Another representative case is the arrival of large language models (LLMs) like GPT-4 by OpenAI, which are increasingly putting pressure on government legislative bodies to regulate their use, which, in turn, will affect organizations, tasks, and agents

Table 2: System factors summary.

Factors to consider	Instructions	Examples

System	• Determine what type of system it is.
	• What is the system's purpose?
	• What is(are) the system's goal(s)?
	  ○ Outcomes
	  ○ Key performance indicators (KPIs)
	• What processes, activities, and tasks there are to fulfil them?
		Type: "An airport / a control tower / a cockpit"
		Purpose: "To achieve safe air travel"
		Goals:
		• Outcomes: "To ensure and maintain sufficient air separation"
		• KPIs: "Throughput, cost, adherence to safety standards, etc."
		"There are standard operating procedures, digital systems to visualize air traffic, radio communications …"

Tasks	Describe the main steps of each task linking the agents and the resources	"The pilot uses system A and resource B to gauge value C and achieve outcome D."
Artifacts	Let's start with the smallest components of the system, the tools used in the system (i.e., physical, and digital). • Who uses them? • For what task? • For what purpose? (Incl. why is that tool used over other alternatives?) Note: Certain tools might be used for multiple tasks in different ways by multiple people!	Radio: • Used by: "Pilot, Air Traffic Controllers" • Used for: "Receiving instructions, requesting information, confirmation …" • Used because: "Reliable and standardized communication channels"
Agents	Then consider any organizations, human operators, artificial agents, and systems that are part of the process considered. • What are their roles? • What are their functions? • What are their responsibilities?	Captain (human) • Co-pilot (human) • Autopilot (artificial) • Air traffic controller (human).

Factors to consider	Instructions	Examples
Processes	It is important to learn what (canonical and exceptional) interactions occur between agents in the system and what purpose they serve. • What information is at the center of those interactions? • Are there any information or action interdependencies between actors? • What interfaces are used to facilitate and support those interactions?	"How actor A performs task X is affected by when and/how actor B performs task Y." • "Plane position between Pilot and Air Traffic Controller." • "The pilot needs the Air Traffic Controller's permission to take off (sequential interdependence)." • "Verbal communication via radio and graphical user interface X."
	Context. Where the first part of this step focuses on what is included *in* the system, the second part shifts attention to what could affect the system from the *outside*. • Which external or contextual factors can impact the system's general function and performance?	• Weather conditions • Wars • Market conditions • Technology advancements
Organization	**Stakeholders**. Identify those parties internal or external to the target system that have an interest in project outcomes. It's important to be aware of their difference in stakes and priorities in term of: • Project goals • Processes • Outcomes	• Upper management wants to reduce the flight's duration for cost efficiency • Operative level wants to reduce the pilot's fatigue for safety purposes
Regulations	Regulatory constraints and opportunities can be included here.	

4.1.2 Three iterative phases

When incorporating an idea into a task and its system, or if an entirely new system is designed, we propose three analytical and developmental phases, see Figure 9. The first phase is the activity modelling phase, *modeling key system factors and interactions*, which serves to develop an understanding of the current system's components and its contextual factors. Understanding the system and the context in which it exists is critical to avoid introducing changes that may have a negative impact on safety aspects and to understand the needs of potential users.

The second phase, *designing for safe human-autonomy teaming*, is focused on deciding which task to implement and how it should be implemented, i.e., how it should be integrated into the

system. The effects of the chosen allocation should be carefully evaluated to avoid possible issues such as decreased situation awareness, work overload, etc.

The third phase is *assessing the design proposal for collaboration issues and risks*. The goal of this phase is to assess whether the new design introduces any risks with regards to human-autonomy teaming. Identified risks are either highlighted to be addressed as the system, function or interactions are detailed further later on in the design process, or require iterations of the three phases, especially focusing on the last two.

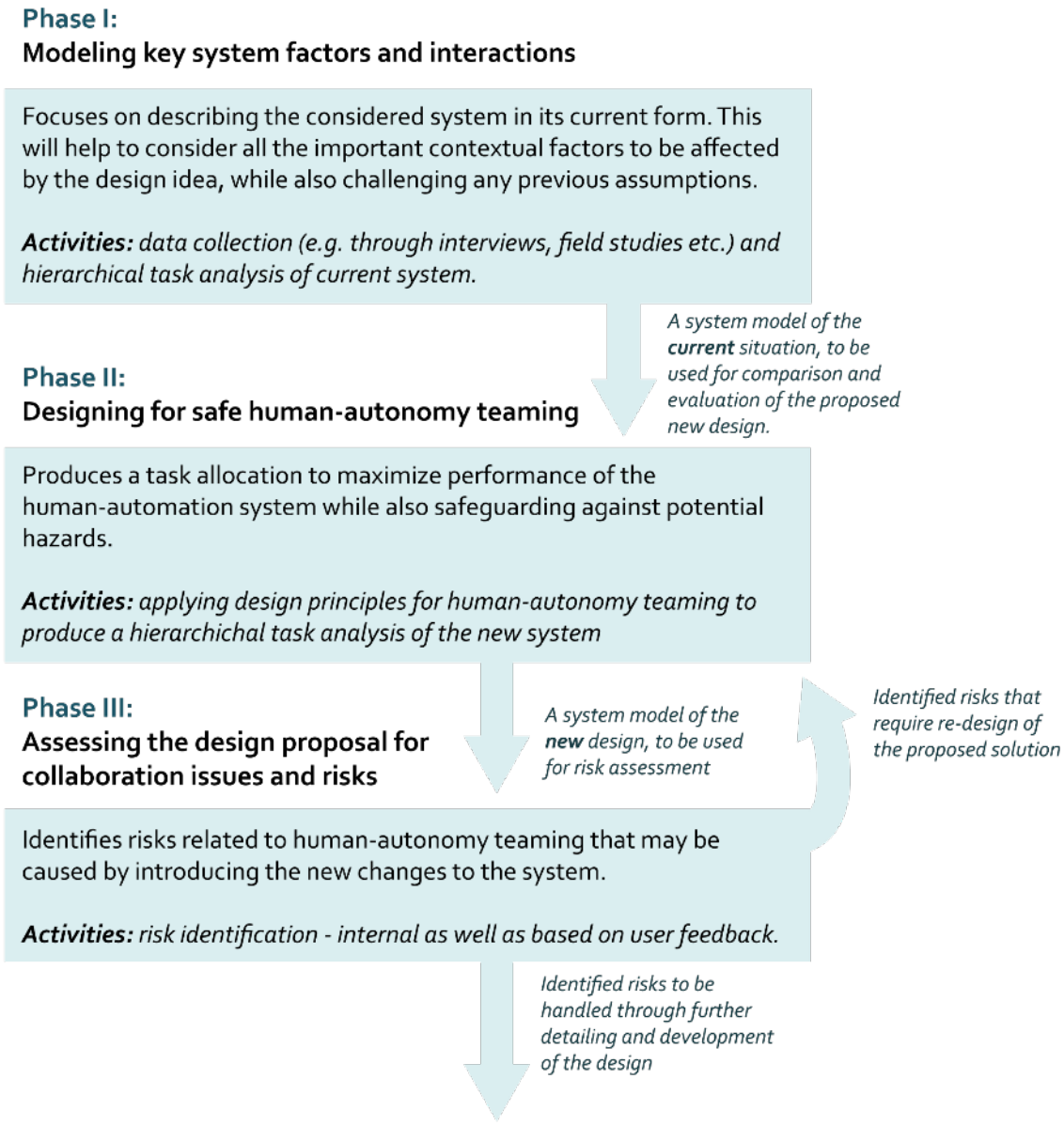


Figure 10. This framework can serve as a guide for how designing automation systems can be approached.

The framework describes what to consider throughout a development process to facilitate effective human-automation interaction and teaming. Applying this model can—from a human factors and usability perspective—increase the quality of a developed system or feature as it

cultivates an awareness of these issues. However, it is not a complete design guide; it is a *general* framework. As such, it makes no assumptions about available resources, competencies, or technologies. The framework can be applied as loosely or thoroughly as required. It is meant to raise awareness, challenge assumptions, and inspire new ideas. How our model is applied will vary between organizations and people depending on their specific domain, expertise, and resources (e.g., time). To summarize; use this framework for inspiration and reflect upon how it is applicable to your domain and design problem.

4.2 Phase I: Modeling key system factors and interactions

The first phase of the SafeTeam framework begins by describing the current work activities of the considered system. To identify and delimit these, consider which system factors are central to the design idea in question – which processes, agents, artifacts, and tasks must necessarily be modelled and understood to develop and evaluate the design? It is important to strike a balance between sufficient coverage and inclusion of relevant system factors on the one hand, and manageable model complexity on the other. In short, the model should be a simplified description of the system activities, not as vast and complex as the system itself. Iterating the model can ensure that contextual factors that affect or are affected by the design proposal are included and considered through the remainder of the design process. The model of the current system activities created in this phase will support subsequent design and assessment phases.

4.2.1 Purpose

Introducing a system, task or a tool in a work environment should reflect the intended or desired change outcomes. As this phase will challenge any previous assumptions about the system and its components, it is an important tool to avoid bad designs or implementations based on incorrect information. By understanding the system's overall architecture, interactions, contexts, and its purpose, one can more accurately evaluate the design motivation.

The model created in this phase can provide a common ground for an array of stakeholders, from executives to implementors and end users. This can help identify design shortcomings and risks while also generate improvement or optimization suggestions. By breaking down tasks into smaller components, you can gain a better understanding of how a system or process works and identify areas where changes could improve efficiency or effectiveness. Understanding the impact of, or effect on contextual factors at an early stage of design can help prevent out-of-the-loop phenomenon, decrease development costs, and result in safer and more competitive system and product designs.

4.2.2 Input

The modelling is initiated by an idea; a design proposal or change requirement prompted by inefficient task processes, inadequate artefacts etc. The design idea guides and focuses the work; data collection about the target domain is centered around the idea, moving outwards to include the relevant domain components (e.g., other tasks, agents and artifacts). Appropriate data collection methods include interviews, focus groups, field studies, contextual inquiries, cognitive walkthroughs etc. A guide on conducting interviews is found in appendix A. Advices presented for conducting interviews can be found useful for other data collection methods as well. Furthermore, the design team's own domain knowledge could also contribute data for the system modelling phase.

4.2.3 Instructions

To integrate all the information collected into an effective tool, we guide you through the construction of a *Hierarchical Task Analysis* (HTA) of the current system. Hierarchical Task Analysis is typically used when modelling activities and the aim is to describe and compare the different tasks that an operator must perform to meet a predefined objective (Annett, 2004). This method is based on numerous models that allow decomposition of tasks into subtasks, the relationships (sequential, parallel, or alternative) between subtasks, and sometimes even the tools needed to accomplish the task and meet the objectives. HTA is a deductive (top-down) decomposition method for which the level of decomposition (of a task into subtasks, themselves decomposed into sub-subtasks etc.) is often unknown a priori. The necessary level of detail is usually only known when the HTA is performed, which may require several iterations with the operators to identify the relevant sub-task level to stop at.

HTA is a useful tool for understanding complex systems or processes by breaking them down into smaller, more manageable components. It involves creating a hierarchical structure of tasks where each task is broken down into sub-tasks, and each sub-task is broken down further into even smaller sub-tasks, until you reach a level where the tasks are simple enough to be easily understood. You will visually map out different factors—tasks, artifacts, agents, and processes—their properties, and dependencies required to achieve your system's goals.

HTA step-by-step

Below is a list explaining how to build your HTA. Below the list you will also find all components in the HTA explained together with examples.

1. List the important system factors based on the idea (read more about system factors in the framework introduction).
2. List the involved roles and assign each role a color.
3. Define high-level tasks.
4. Elaborate the tasks related to the idea with subtasks. A task can only be elaborated if there are *at least* two sub-tasks.
5. Add plans; holding information on how tasks are related to each other, e.g., if tasks are carried out in parallel or in sequence.
6. Tag tasks with task type: information acquisition, information analysis, decision selection or action implementation.

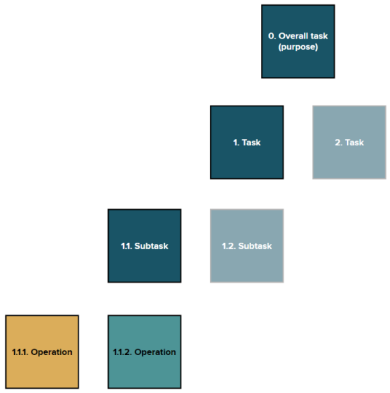
Tips

A whiteboard, online whiteboard tool or PowerPoint (SmartArt - Hierarchy) are suggested tools for building your HTA.

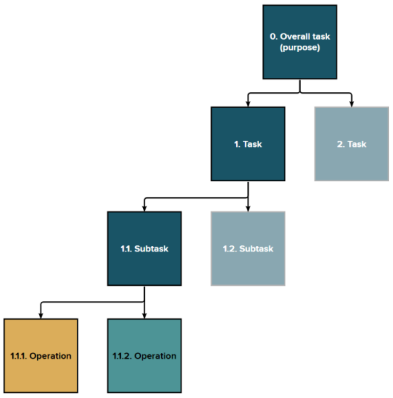
A task can be tagged with more than one task type, but doing so is an indication that the task might be possible to detail further - decide if this is necessary based on the scope of the idea.

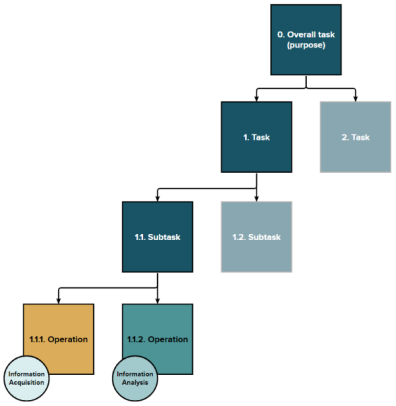
Table 3 Explanation of SafeTeam HTA components

HTA components and information	Examples

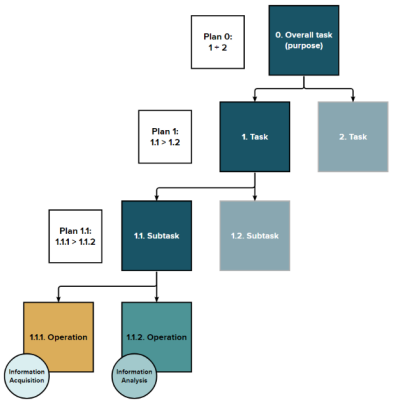
• Task boxes:
  ○ Tasks
  ○ Subtasks
  ○ Operations (leaf nodes (bottom nodes); what agents *do*)
• Numbers to indicate hierarchy of tasks. The numbering also keeps track of the hierarchy when tasks are transferred to the tabular format later on.
• A task must have at least two subtasks, otherwise the single subtask is moved up and replaces its parent task.
	

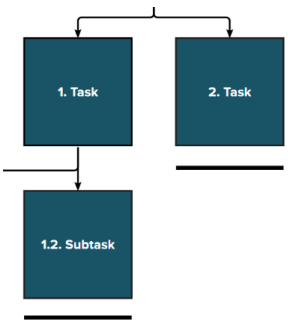
• Color coding for agents.
  ○ Provides quick overview.
  ○ Use as many colors as necessary to describe and visually differentiate your system agents.
	• **Tasks carried out by multiple agents.**
	• Human
	• Artificial agent

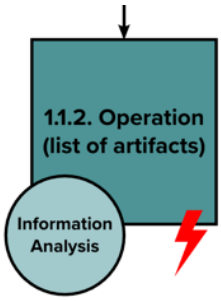
• Connecting lines to highlight hierarchical relationship between tasks, subtasks, and operations.
  ○ Only between hierarchical levels.
  ○ Never between tasks at the same hierarchical level.
	

• Task type indicator. Classify each task (focusing on the leaf nodes) with its proper task type:
  ○ Information acquisition: 'sensing and registration of input data'.
  ○ Information analysis: 'prediction, integration (combination of various values into one), ...'.
  ○ Decision and action selection: 'decision from among decision alternatives. Propose a course of action'.
  ○ Action implementation: 'execute the action chosen'.
	

HTA components and information	Examples

• Plans showing order or conditions for tasks. Can be expressed as:
  ○ Linear (then, >, →)
  ▪ 1 then 2 then 3 then 4
  ▪ 1>2>3>4
  ○ Non-linear (or, /)
  ▪ 1/2/3/4
  ▪ Do 1, 2, 3, 4 in any order
  ○ Simultaneous (and, +, &)
  ▪ 1+2+3+4
  ▪ 1 and 2 and 3 and 4
  ▪ Do 1, 2, 3, 4 at the same time
  ○ Conditional (if *condition* then, X? >)
  ▪ X? Y>1 N>2
  ▪ If X, then 1 else 2
  ▪ Do as required
  ○ Cyclical
  ▪ 1>2>3>1…
  ▪ Repeat the following until
  ○ Selection (any of, :)
  ▪ 1:2:3:4
  ▪ 1 or 2 or 3 or 4
  ▪ Choose one of the following
	

• Bottom line to signal that a leaf will not be expanded further. • Maximum 4-5 levels of subtasks This is of importance for ease of comprehension. If five levels are not sufficient to describe the system, consider separating your general HTA structure into several trees (e.g., one structure per subtask).	

HTA components and information	Examples
• Insert information on relevant artifacts as a list in the leaf nodes.	
• Risk tags to indicate subtasks and operations with known risks. The purpose of the risk tags is to visually indicate that these are tasks that should be given extra attention when comparing the new design to the current system.	

This process is highly adjustable to your systems and should be considered as a set of optimal items to be selected depending on your needs. You may need several iterations of factors framing and HTA structuration to obtain the right focus: widen up or narrow down your system; considered additional factors; etc. However, this will not wipe out but rather complete the work already performed.

It is not necessarily important to finish the entire model of the current system before moving on to modelling the idea (see Phase II). When modelling the idea, one can find what tasks are necessary to expand further and what tasks to leave at a high level.

4.2.4 Output

The system model of the current situation will be used for comparison and evaluation of the proposed new system.

4.3 Phase II: Designing for safe human-autonomy teaming

Any fast-paced operational domain may benefit from the introduction of automation, but automating tasks without fully assessing the consequences may be hazardous. This chapter contains an approach where automation solutions are proposed and structured in a way that supports later evaluation and detailing. It is an iterative approach and potential hazards that are identified may require the designer to re-assess aspects of human-automation interaction and modify the task allocation from the initial idea.

A joint activity is a set of actions carried out by an ensemble of partners, human or non-human. All participants must enter into an agreement, be mutually predictable, be mutually directable and

maintain common ground, among other things. They must be able to rely on each other when considering their own actions, which can facilitate synchronized actions and efficiency.

To create a safe autonomous system from a human-centered perspective we propose a set of design principles listed below. Keep these in mind while suggesting system requirements to mitigate any risks that was identified and remember: These are suggestions not truths.

4.3.1 Purpose

The main purpose of this step is to produce a task allocation that maximizes performance of the human-automation system while also safeguarding against potential hazards. Like for Phase I, it is highly recommended that the work in Phase II is done collaboratively; the goal of the HTA is to facilitate discussion. Therefore, if possible, include end users in the design process (i.e., a participatory design approach). If not, collect their feedback on the design proposal at a later stage, and iterate.

4.3.2 Input

The HTA produced in the previous phase will constitute the starting point for the work performed in phase II.

4.3.3 Instructions

During phase II the HTA created in Phase I will be adjusted to reflect the idea that initiated the design process. To assist with the task allocation, use the human-autonomy design principles below and create a HTA reflecting the proposed design. Once the new HTA is created, the impacted leaf nodes of the HTA are transferred to a table. The tabular format allows for recording additional aspects that were not included in the HTA (Annett, 2004). The table proposed in the SafeTeam Framework suggests including details of the tasks that will later help addressing risks and specifying the design considerations for the system changes.

4.3.3.1 Human-autonomy design principles

To create safe autonomous systems from a human-centered perspective we propose a set of design principles listed below. It is important to emphasize that these are recommendations and designed to provide new insights and inspire new areas of inquiry.

Foundation of collaboration

1. **Agents should share a common goal.**

   Any agent participating in a task should understand and accept the common goals. In relation to the common goal, any individual agent participating in the effort should be able to represent, reason about, and modify their individual goals to ensure coordination and maintenance of the common goals.

   Considerations:

   - **Clear Goal Definition:** The common goal should be clearly defined, as well as each individual goal. Ambiguity may lead to confusion or inefficiency.
   - **Goal negotiation**: If agents can reason about and modify their goals, they will be able to adapt when situation changes. Develop protocols to maintain team cohesion and resolve potential conflicts.

2. **Agents should be able to share their status and intentions and interpret and observe the intentions of others.**

   Agents should be able to provide real-time updates on status, capacities, actions, decisions, and intensions to enable other parties to maintain situation awareness and promote making informed decisions. It is equally important to represent this information in a manner that allows for teaming agents to observe and interpret it to evaluate and detect possible failures.

   Considerations:

   - **Interfaces, message formats and protocols:** Communicating intent and the ability to interpret the information should be dependent on a standardized information flow. Consider the transparency of the system by sharing why decisions are made, highlight data and algorithms to offer insights into how the system interprets inputs to produce outputs.
   - **Intention sharing:** Agents should share their intentions as it enhances coordination, trust, and synergies among team members which lead to more efficient collaborative efforts.
   - **Status updates**: Agents should periodically send updates on their current state, tasks they are working on, and any changes in their capabilities or limitations. Changes and event should be highlighted to ensure that teaming agents are aware of changed conditions (Christoffersen & Woods, 2002).

3. **Agents should be directable.**

   The human operator should be able to guide, control and influence the autonomous system to mitigate risks related to unforeseen or complex situations. These directions might be explicit e.g., task allocation or direct assignments but may as well be more subtle e.g., providing information (Johnson et al. 2014).

   Considerations:

   - **User interface design:** The user interface should be clear and allow human operators to easily understand and interact with the autonomous systems. Consider intuitive controls, informative displays, and a well-organized information flow.
   - **Override capabilities:** Consider whether the operator should be able to intervene or adjust the system's behavior in real-time and the potential ripple effects.
   - **Ethical and safety constraints:** The system should have limits and constraints within itself to prevent it from taking actions that are unsafe or unethical. Even in an autonomous mode the system should operate within predefined boundaries.

Cognitive Load

4. **Designers of the system should strive for Shared Situation Awareness.**

   Agents should be able to understand the dynamic environment they are operating in and the elements it is containing, the comprehension of their meaning, and the projection of their status in the near future (Parasuraman, R., Sheridan, T.B., and Wickens, C.D., 2008).

   Considerations:

   - **Common ground:** Consider how to achieve individual situation awareness and how to build a shared situation awareness by sharing knowledge, beliefs, and assumptions.
   - **Visual Scanning**: The operator should be able to seek (additional) information when needed. Attention guidance, such as alerts or notifications might be deployed to aid.
   - **Integration of Knowledge:** Consider how knowledge can be split into smaller parts through display integration.

5. The system should enable optimal Mental Workload.

When designing a system, the distribution of mental effort, attention, and cognitive resources should be considered. It is important to analyze the cognitive and perceptual demands placed on the human operator and their abilities (Parasuraman, R., Sheridan, T.B., and Wickens, C.D., 2008).

Considerations:

- **Optimal workload:** A high workload may cause mental fatigue and decreased performance, while a low mental workload may on the other hand lead to reduced focus and decreased performance. The workload and task complexity should allow agents to adapt to the cognitive capabilities and limitations of human operators, optimizing the allocation of tasks and information to ensure a stimulating mental workload.
- **Task complexity:** The level of decision-making, number of variables and the novelty of the task will affect the mental workload. Consider these parameters when designing tasks.
- **Information Load:** Consider the amount of data and manner in which data is presented through screens or other sensory inputs.
- **Interruptions and multitasking:** How often human operators are interrupted, or the number of concurrent tasks affect the mental workload.

Trust and system Reliability

6. The system should foster mutual trust.

Building trust between team members in a human-autonomy system is an essential factor to consider since trust directly influences how a system tends to be used. Ensuring trustworthiness in autonomous systems not only enhances their acceptance of the system but also mitigates the risk of misuse or underuse (Lee, J.D., & See, K.A. 2004). If the human operator has too low level of trust in an automated system, they may disregard information provided by the system. However too high level of trust might cause the operator to fail to monitor important information (Bisantz, A.M. and Seong, Y., 2001).

Considerations:

- **Reliability**: Is a fundamental factor when considering trust in a system. The system should be able to perform its intended functions consistently and accurately over time (Bisantz, A.M. and Seong, Y., 2001).
- **Transparency and understandability**: Automated systems being transparent with their intentions, explaining what information is accounted for, and how it makes decisions improve trust from human operators. Consider how the results of algorithms are displayed to the operator to ensure that no errors are introduced due to bad design or inaccurate results. The design should aid decision-making and enhance system monitoring.
- **Familiarity:** When an operator recognizes the system components it will likely enhance trust, thus it is helpful to consider how to design a system that the operator is familiar with.
- **Error Handling and Recovery:** The type and frequency of errors can affect the operators trust in the system e.g., too many false alarms and false positives. A system's ability to recover and maintain performance can also instill trust.

7. Agents should act in compliance with ethical standards.

Autonomous agents involved in collaborative tasks should adhere to a shared set of ethical standards. It is important that agents understand and accept these ethical principles to ensure ethical decision-making and behavior within the joint human-autonomy team.

Considerations:

- **Clear Ethical Standards:** Define the shared ethical standards to avoid ambiguity or misinterpretation. Ensure that all agents have a common understanding of ethical behavior (Awad et al., 2018).
- **Ethical Decision Support:** Equip agents with the capability to make ethical decisions, guided by the shared ethical standards, to foster ethical behavior during task performance (Vanderelst & Winfield, 2018).
- **Ethical Accountability:** Establish mechanisms for monitoring and evaluating agents' compliance with ethical standards, allowing for accountability in case of ethical violations (De Graaf, 2016).
- **Ethical Transparency:** Promote ethical transparency in ethical decision-making processes. Enable human team members to question, understand, and trust the ethical choices made by artificial agents.
- **Conflict Resolution:** Implement procedures for resolving situations where ethical conflicts arise within the human-autonomy team. Ensure that ethical disagreements are addressed in an ethical and collaborative way.
- **Dynamic Ethics:** Consider protocols for enabling agents to adapt their behavior when shared ethical standards evolve or in response to emerging ethical dilemmas (Bonnefon et al., 2016; Lin, 2016).

4.3.3.2 Hierarchical Task Analysis (HTA) and Tabular Task Analysis (TTA) - future system

A new HTA is produced to reflect the design idea. As previously mentioned, as the initial idea is modelled, one may find that tasks in the previous HTA must be further expanded and iterations back to Phase I are encouraged.

1. The SafeTeam method for task allocation takes the HTA model from Phase I as its starting point. If working in a digital format, simply copy and paste the HTA from Phase I to get started. If you are working with a physical version, e.g., whiteboard or tabletop artifacts, ensure that the original HTA is thoroughly documented (e.g., photographs) or, preferably, begin working with a second whiteboard or tabletop setting.

2. In a workshop setting (with colleagues or with prospective end users), edit the tasks (nodes/branches) and operations (leaves) to represent your design idea.

    a. Use the appropriate color coding to represent which agents perform which operations. Is a manual (human) operation replaced by a digital system? Change the operation card color (same operation, different agent).

    b. Add and remove tasks/operations as needed. Adjust the artifacts descriptions.

    c. Decide whether to break tasks into their operations or whether to "black box" them (drawing a line under their task card).

    d. Adjust the tasks/operation numbering.

    e. Refine the plans; add plans to any new tasks, adjust their numbers.

A note about levels of automation: the levels (see section 2.4) can be represented in the HTA by breaking tasks down to a level where the HTA clearly shows how artificial and human agents interact, e.g., "system does X, then Y, then requests input from human, then human accepts/rejects suggestion, then system does Z…" etc. If a design solution concerns adaptive automation, multiple HTA versions may be required to fully model the levels of automation involved.

Task allocation: discussion topics

In the next phase more attention will be given to identifying risks and **design** considerations (for future development). Some risks might be too severe or unsuitable to address through **design** considerations and may require adjustments in the task allocation. To minimize risk of rework, the following questions may be discussed as the task allocation is defined:

1. For the affected human agent(s), how does the design impact their overall workload?
2. Compare the task types left for the human agents to perform.
    a. Does the design imply monotone or varied work?
    b. Consider the effects on work satisfaction.
3. How may subsequent tasks/operations be affected in terms of cognitive demands? For instance, if the information analysis task preceding a human decision is replaced by a digital analysis system, how is the human decision task impacted? If the prerequisite analysis work is no longer done, how can human decision support be provided instead?
4. Consider the "soft" or "indirect" task dependencies (e.g., cognitive synergies between tasks).
    a. Were any positive (e.g., situation-awareness-supporting) dependencies eliminated due to the design change?
    b. Were any new positive dependencies introduced because of the design change?
5. For a chosen allocation, what are the envisioned consequences at…
    a. The system level - Effects on related functions in the system (e.g., undermining the supply of information for another task, or resulting change requirements in related functions, e.g., new functionality/output required).
    b. The organization level - does the change affect the way work is organized (role allocation, authority, responsibilities, team composition, etc.)
    c. The task level - Effects on communication, workflow, work sharing, shared situation awareness…
    d. The individual level - Effects on workload, attention, memory, situation awareness, systems understanding, knowledge, work satisfaction, etc.
6. How might these consequences affect performance (e.g., selected KPI's)? Or more general KPI's?

When the proposed task allocation has "matured" and settled in HTA form, it should be evaluated in tabular form using the TTA. It is important to understand the influence of a chosen allocation on a contextual level.

1. Transfer the operations (leaf nodes) to the task table. Keep the agent color coding. Overarching tasks may also be added in a column for overview purposes.
2. Classify the task types: Information acquisition, information analysis, decision selection or action implementation. Consider color coding for quick overview.
3. Mark all new or changed (e.g., changed agent) tasks.
4. Write the dependencies for all new tasks (e.g., data input or output, cognitive dependencies that users can experience or other reasons why two tasks may impact the performance of each other).
5. Add the identified artifacts.
6. Mark the tasks that are dependent on the newly added task.

7. If necessary, add any comments to the operations.

Table 4. Example of a TTA, selected tasks of a system, only including new and affected tasks

Task	Operation	Task type	New task (new), affected by new task (affected) or task that changed agent (new agent)	Artifacts	Task dependencies	Comment
1 Plan ahead	1.1 Collect approach-data	Information acquisition	New	ML artifact		
	1.2 Predict unstable approach	Information analysis	New	HMI	Affected by: 3.2, 3.3, 6,7 Affecting: 2, 3.1, 3.2, 5.2.1, 5.2.2	
3.1 Monitoring aircraft states	3.1.1.2 Monitor aircraft speed data	Information analysis	Affected	HMI	Affected by: 1.2, 3.1.1.1 Affecting: 2, 3.2, 5.2.1, 5.2.2.	
	3.1.2.2 Monitor track	Information analysis	Affected	HMI	Affected by: 1.2, 3.1.2.1 Affecting: 2.1, 3.2, 5.2.1, 5.2.2	
	3.1.3.2 Monitor vertical track data	Information analysis	Affected	HMI	Affected by: 1.2, 3.1.3.1 Affecting: 2.1, 3.2, 5.2.1, 5.2.2	
5.2 Intra-Cockpit	5.2.1 Announce plan to other pilot	Action implementation	Affected		1, 4, 6	
	5.2.2 Announce deviations	Action implementation	Affected	HMI e.g., PFD	3.1	

Color digital assistant	Color indicating both pilots	Color indicating Auto Pilot/Pilot Flying	Color indicating Pilot Monitoring	Color indicating ATC

4.3.4 Output

The output of Phase II comes in two parts:

1. The revised HTA, visually describing the system and task allocations as envisioned for the new or changed system.
2. The TTA table describing the envisioned design, prepared for evaluating its properties in terms of risks related to human-autonomy teaming. The TTA will be further developed in Phase III, where risks will be identified and used to define considerations for future work.

4.4 Phase III: Assessing the design proposal for collaboration issues and risks

In this phase, HAT-related risks of introducing new changes or implementing a new system in a certain way will be established and design considerations to mitigate said risk will be created. It is encouraged to discuss risks in a multidisciplinary group – adding many different perspectives. Like the previous steps, the work performed during this phase is iterative and might result in re-assessment of the task allocation performed during phase II.

4.4.1 Purpose

The purpose of this phase is to explore how different task allocations may lead to certain risks and generate design requirements to mitigate these risks, focusing on Human-Autonomy aspects. The list that is produced in this step is the basis for further development and detailing of the design.

4.4.2 Input

The TTA that was created in the previous step is used and further detailed in this phase.

4.4.3 Instructions

The TTA created in phase II will now be expanded with two additional columns:

- Risks
- Design considerations to mitigate identified risks

4.4.3.1 Future system TTA – risk assessment

Add a column to the TTA produced in Phase II. This column will contain risks that are associated with each operation listed in the TTA. If possible, return to users or other stakeholders that you interacted with in Phase I and collect possible risks from their points of view as well. Additionally, use the list of sample questions below for inspiration to identify potential risks. Note that the questions are categorized by task type. Focus on the task type questions that are relevant to your TTA items. Also note that the questions do not differentiate between human and artificial agents; the topics to be considered are relevant in both cases.

The question list below uses the words *systems* and *agents.* Keep in mind that *a system* can be a *system of systems* and it might be helpful to zoom out from subsystems. *Agents* refer to both *human agents* and *artificial agents*.

Information Acquisition

1. Data Collection and Accuracy:

- Under what circumstances may the data collected from the environment be incomplete or when is the accuracy and reliability of this data is not ensured?
- To what extent is it possible that faulty, outdated, or unreliable information is handled as true? What are the consequences of such incidents?

2. Adaptation to Changing Conditions:

- In what situations could agents fail to adapt to changes in their environment or to new types of information that they need to collect?
- When could the system fail to adequately support agents in challenging data collection scenarios?

3. Data Submission, Consistency, and Presentation:

- Are there specific circumstances that might hinder timely submission of information, and what could be the consequences of such delays?
- What could cause data inconsistencies, conflicts when merging data from multiple sources, potential misinterpretation, or information overload during data presentation?

4. Feedback and Learning:

- What could be the consequences of inadequate feedback regarding collected data, e.g., issues of format or quality?
- What risks could arise from agents learning (or not) from their actions or improving their data collection methods?

5. Ethical and Privacy Considerations:

- Do agents handle sensitive or confidential information, and what risks are associated with the handling of such information?
- Can you think of potential ethical dilemmas or privacy breaches that could occur when collecting information?

Information Analysis

1. Analysis Techniques and Error Handling:

- What techniques (tools, methods, algorithms) will agents use to identify patterns and anomalies, and what skills and expertise are required for effective analysis? When may this be lacking and what are the consequences?
- What contextual factors could cause misinterpretation, ambiguity in analyzed data, or faulty error detection/correction in the analysis process?

2. Adaptation to Changing Conditions:

- What could cause the data conditions to change over time? What risks could that introduce to the analysis?

3. Feedback and Learning:

- What could be the consequences of inadequate feedback regarding analyzed data?
- What risks could arise from agents learning (or not) from their actions and interactions with others to improve their data analysis methods?

4. Communication and Collaboration:

- How will agents collaborate, share insights and interpretations, during the analysis process? In what situations could this have a negative impact on the analysis process?
- Is there any risk of misunderstanding of agents' roles, responsibilities, or accountabilities? What could be the consequences of such misunderstandings?

5. Managing Information Overload:

- In what situations could there be an overwhelming amount of information for agents to process, and what risks could that introduce?

6. Ethical Considerations:

- Are there risks that potential biases or ethical concerns have an impact on the analysis process?
- What could be the implications of biased or unethical analyses?

Decision Selection

1. Ethical Considerations:

- Are there risks that agents make decisions or recommendations in contradiction with ethical standards and human values, and what could be the consequences?

2. Adaptation to Changing Conditions:

- In what situations could changing environmental conditions or new information negatively impact the decision-making process?
- What could be the consequences if agents cannot monitor and reevaluate decisions based on new information?

3. Feedback and Learning:

- What could be the consequences of inadequate feedback regarding options provided or decisions made?
- What risks could arise from agents learning (or not) from their actions and interactions with others to improve their decision-making skills?

4. Collaboration, Communication, and Clarification:

- Can agents collaborate, share insights, and reach consensus, during the decision-making process? What could cause this collaboration to fail, and what could be the consequences?
- For what reasons could agents need to seek clarification or feedback before finalizing decisions? What risks could arise if agents cannot clearly convey their decisions and rationale?

5. Decision Criteria, Prioritization, and Complexity:

- Are there risks that agents fail to consider necessary criteria or methods to compare decision recommendations, or to prioritize different factors? Especially looking at complex scenarios requiring analysis of multiple factors or situations prone to cause decision paralysis or information overload.

6. Risk Assessment and Mitigation:

- Are there potential high-risk scenarios where agents might fail to assess and mitigate potential risks associated with its decision recommendations?

7. Human Override and Intervention:

- What situations might be reason for human operators to override or modify artificial agents decision recommendations? What procedures must be in place to allow this?
- What might be the consequences if there are conflicts between human decision and agent recommendations?

8. Decision Confidence, Reliability, and Data Access:

- How does the agent communicate its confidence level or uncertainty in its decision recommendations? What are the consequences if the agent's confidence level or uncertainty in its decision recommendations are not clear?

Action Implementation

1. Adaptation to Changing Conditions:

- What changing conditions may be reasons for modifying an agent's action? What are the consequences if this is not possible?

2. Feedback and Learning:

- How does the system provide feedback about the status and results of actions implemented by other agents, and what mechanisms are in place to promptly alert agents in case of anomalies or unexpected outcomes?

3. Execution Algorithms and Procedures:

- What methods (guidelines, procedures, algorithms) will the agent use to execute actions based on decisions? Is it possible that these methods do not ensure accurate or reliable execution of actions?
- What issues might be associated with the methods (e.g., guidelines, procedures, algorithms) used by the agent to execute actions based on decision, and is there a risk that these methods may not guarantee accurate or reliable execution of actions?

4. Real-Time Responsiveness:

- What risks are associated with the speed at which an artificial agent implements actions once a decision is made? What would be the consequences if this implementation is delayed?
- What are the potential sources of latency or delays in the execution of actions, and what are the consequences?

5. Error Detection and Recovery:

- What are the consequences of errors, deviations, or failures in the agent's action implementation?
- Are there mechanisms for the agent to recover from errors and resume operations?

6. Action Integration, Conflict Prevention, and Collaboration

- What issues are associated with the integration of actions performed by agents in a system with ongoing activities and processes carried out by other agents?
- How can the system prevent conflicts or disruptions arising from agent actions, and can agents seek approval or guidance from other agents if needed?

Spend some time to discuss and reflect over the identified risks. Is there a common reason for several risks? Are there patterns or chain reactions? By doing this you might be able to decide whether to return to the function allocation in Phase II and make changes there to avoid risks, or if you should minimize the risks through more specific design considerations.

4.4.3.2 Design considerations

Add another column intended for design considerations that might mitigate the identified risks. It might not be possible to propose design aspects that will minimize the risk, in which case it might be necessary to consider a new function allocation that that will lead to the risk being avoided or better handled. Keep the design principles in section phase II in mind to find solutions and to avoid introducing any new risks when proposing design considerations.

4.4.4 TTA example

The following table is the example of a TTA from phase II, with two new columns are added according to the instructions.

Table 5 Expansion of table 4 with added columns Risks and Design considerations.

Task	Operation	Task type	New task, affected by new task *or* task that changed agent	Artifacts	Task dependencies	Risks	Design considerations	Comment
1 Plan ahead	1.1 Collect approach-data	Information acquisition	New	ML artifact				
	1.2 Predict unstable approach	Information analysis	New	HMI	Affected by: 3.2, 3.3, 6,7 Affecting: 2, 3.1, 3.2, 5.2.1, 5.2.2	False predictions False positive - unnecessary GA → introduces another risk False negative - suggests a stable approach even though it might be(come) unstable, situation perceived as safe even though there is an inherent risk	To be decided	
3.1 Monitoring aircraft states	3.1.1.2 Monitor aircraft speed data	Information analysis	Affected	HMI	Affected by: 1.2, 3.1.1.1 Affecting: 2, 3.2, 5.2.1, 5.2.2.	Complacency of Pilot Monitoring, Too strict UA limits will lead to nuisance alerts, Too strict UA limits will lead to unnecessary GA,	Recurring pilot training Appropriate setting of alert limits	
	3.1.2.2 Monitor track	Information analysis	Affected	HMI	Affected by: 1.2, 3.1.2.1 Affecting: 2.1, 3.2, 5.2.1, 5.2.2	Complacency of Pilot Monitoring Too strict UA limits will lead to nuisance alerts, Too strict UA limits will lead to unnecessary GA,	Recurring pilot training Appropriate setting of alert limits	
	3.1.3.2 Monitor vertical track data	Information analysis	Affected	HMI	Affected by: 1.2, 3.1.3.1 Affecting: 2.1, 3.2, 5.2.1, 5.2.2	Complacency of Pilot Monitoring Too strict UA limits will lead to nuisance alerts,	Recurring pilot training	

Task	Operation	Task type	New task, affected by new task *or* task that changed agent	Artifacts	Task dependencies	Risks	Design considerations	Comment
						Too strict UA limits will lead to unnecessary GA,	Appropriate setting of alert limits	
5.2 Intra-Cockpit	5.2.1 Announce plan to other pilot	Action implementation	Affected		1, 4, 6	Complacency of Pilot Monitoring Misleading, unclear communication	Training Use of standard terminology	
	5.2.2 Announce deviations	Action implementation	Affected	HMI e.g., PFD	3.1	Complacency of Pilot Monitoring Too strict UA limits will lead to nuisance alerts, Too strict UA limits will lead to unnecessary GA,	Training Use of standard terminology	

Color digital assistant	Color indicating both pilots	Color indicating Auto Pilot/Pilot Flying	Color indicating Pilot Monitoring	Color indicating ATC

4.4.5 Output

The TTA with tasks, subtasks, risks, design proposals that is evaluated to a set of design principles, could be used a foundation for the development requirements for the UX designers and developers.

5 Discussion

In this report, we propose a methodological framework to support the consideration of human factors constraints related to human-autonomy teaming which is a fundamental requirement when designing systems that combine artificial and human agents. The framework provides 1) a way to systematically model tasks in sociotechnical systems, 2) tools for identifying coordination/cooperation problems generated by the introduction of an artificial agent into a sociotechnical system, and 3) design principles to compensate for these problems.

The proposed framework takes inspiration from multiple established methods, primarily HTA (Annett, 2004), coactive design (Johnson et al., 2014), and types and levels of autonomy (Parasuraman et al., 2000). Centering the framework around HTA activities offers a collaborative and visual way of modelling the tasks and procedures of the target system. This format has the added benefit of being similar to flow charts and diagrams, potentially increasing its initial familiarity for readers with a technical background. As a method, HTA has a proven track record in terms of both task analysis and task design (REF). Its flexibility allowed us to introduce and incorporate the task type taxonomy by Parasuraman et al. to provide some structure to the function allocation (i.e., task design) and risk assessment phases. Aspects of coactive design regarding agent (or task) interdependencies, observability, predictability, and directability were adapted and included in our framework in the form of design guidelines and risk assessment topics/questions. Although coactive design builds upon an initial HTA, rather than addressing these HAT issues after the fact, we opted to include the coactive-design-inspired HAT guidelines into the HTA procedure itself (phase II), with the intention that function allocation using HTA be done collaboratively, creatively, iteratively, and reflectively. All together, we believe that the proposed framework can help non-experts in human factors to consider relevant human factors and HAT aspects that were previously unaccounted for due to lack of expertise or resources.

It is important to stress again what this framework *is*, what it is *for*, what it *does*, and what it *does not* do. First, the proposed framework is not novel in its content per se, but in its packaging. It is a toolbox with a curated set of tools included. Our intention is for these tools to be useable, flexible, and adaptable to serve multiple (and new) purposes. In fact, we encourage the interested reader to customize the framework to fit their specific needs and design circumstances. Second—the previous point notwithstanding—the framework is, at its core, for doing *function allocation* and *task design* with a particular focus on achieving safe and efficient *collaboration* between human and artificial agents. Third, it does this by providing ways to model and visualizing tasks in current work settings (HTA; phase I), visualizing the potential effects of modifications on agents, tasks, and procedures while guiding the design work toward successful human-autonomy teaming (HTA, design guidelines; phase II), and promoting reflexive assessment and (re)design with respect to issues of collaboration between human and machine (risk assessment questions by task type; phase III). Finally, and critically, the proposed framework is **not** for doing User Interface (UI) design (buttons, icons, colors, and other visual element of technology) or User Experience (UX) design (overall look, feel, and general emotional response of technology use). This framework is for designing who does what, in what manner, and in what order, without saying anything about how the tools designed to perform those tasks should look or feel. As such, the proposed framework does not replace other UI or UX methods or production pipelines currently used by prospective users of our framework. Rather, it serves to complement those pipelines regarding the design of safe and efficient HAT configurations and task allocations. If anything, in that respect, the framework is intended to replace any HABA-MABA-based task design methods in use today.

Naturally, the proposed framework is not perfect, and has some limitations to consider. First, the framework is not exhaustive or comprehensive in a narrow sense, but in a broad sense it covers most of the relevant enablers of HAT, although none of them in-depth. However, this is by design. Established methods—like the ones described in this report—have either a narrow focus or require

considerable experience and resources to apply. Our approach was instead inspired by the Pareto principle which suggests that 80 percent of the consequences come from 20 percent of the causes. It is also sometimes called the "law of the vital few." Applied to the current case, we posited that by using the proposed framework, designers could achieve 80 percent of the desired HAT and collaboration qualities for 20 percent of the effort compared to other established (more complex and labor-intensive) methods. Additionally, by offering a broad toolset, we hope to raise the minimum level of awareness about human factors and HAT in the aviation industry overall. However, if at all possible, we still recommend acquiring the assistance and expertise of human factors experts when designing for HAT.

A second limitation concerns the development of the framework itself. As detailed in the method section, the framework was iteratively developed in collaboration with and through a series of use case applications. As these use cases were ongoing project work packages, the final version of the proposed framework is still untested. There may be an opportunity to test the framework in a third use cast after the delivery of this report. Possible insights and additional detailed framework revisions may be subject to future scientific publication.

6 References

Annett, J. (2004). Hierarchical Task Analysis. In D. Diaper, & N.A. Stanton (Eds.), *The Handbook of Task Analysis for Human-Computer Interaction* (pp. 67-82). Lawrence Erlbaum Associates.

Atmaca, S., Sebanz, N., Prinz, W., & Knoblich, G. (2008). Action co-representation: The joint SNARC effect. *Social neuroscience*, *3*(3-4), 410-420. https://doi.org/10.1080/17470910801900908

Awad., E., Dsouza, S., Kim, R., Schulz, J., Henrich, J., Shariff, A., Bonnefon, J.-F., & Rahwan, I. (2018). The Moral Machine Experiment. *Nature*, *563(7729)*, 59-64. https://doi.org/10.1038/s41586-018-0637-6

Bainbridge, L. (1983). Ironies of automation. *Automatica*, *19(6)*, 775-779. https://doi.org/10.1016/0005-1098(83)90046-8

Berberian, B., Sarrazin, J-C., Le Blaye, P., & Haggard, P. (2012). Automation Technology and Sense of Control: A Window on Human Agency. *Public Library of Science ONE*, *7(3)*, 1-6. https://doi.org/10.1371/journal.pone.0034075

Billings, C.E. (1991). Human–centered aircraft automation: A concept and guidelines. Moffett Field, CA: National Aeronautics and Space Administration, Ames Research Center.

Billings, C.E. (1996). *Human-centred aviation automation: principles and guidelines* (Report No. NASA-TM-110381). National Aeronautics and Space Administration. https://ntrs.nasa.gov/citations/19960016374

Bisantz, A.M., & Seong, Y. (2001). Assessment of operator trust in and utilization of automated decision-aids under different framing conditions. *International Journal of Industrial Ergonomics*, *28*(2), 85-97. https://doi.org/10.1016/S0169-8141(01)00015-4

Bonnefon, J.-F., Shariff, A., & Rahwan, I. (2016). The Social Dilemma of Autonomous Vehicles. *Science*, *352(6293)*, 1573-1576. https:/doi.org/10.1126/science.aaf2654

Bradshaw, J.M., Dignum, V., Jonker, C., & Sierhuis, M. (2012). Human-Agent-Robot Teamwork. *IEEE Intelligent Systems*, *27(2)*, 8-13. https://doi.org/10.1109/MIS.2012.37

Byrne, E.A., & Parasuraman, R. (1996). Psychophysiology and adaptive automation. *Biological psychology*, *42(3)*, 249-268. https://doi.org/10.1016/0301-0511(95)05161-9

Carmody, M.A. & Gluckman, J.P. (1993). Task specific effects of automation and automation failure on performance, workload and situational awareness. In R.S. Jensen & D. Neumeister (Eds.), *Proceedings of the Seventh International Symposium on Aviation Psychology* (pp. 167-171). Department of Aviation, The Ohio State University, Columbus, OH.

Carroll, J.M., Kellog, W.A., & Rosson, M.B. (1991). The Task-Artifact Cycle. In J.M. Carroll (Ed.), *Designing Interaction: Psychology at the Human-Computer Interface* (pp. 74–102). Cambridge University Press.

Chapanis, A. (1965). On the allocation of functions between men and machines. *Occupational Psychology*, *39(1)*, 1–11.

Chen, J.Y., Procci, K., Boyce, M., Wright, J., Garcia, A., & Barnes, M. (2014). Situation Awareness-Based Agent Transparency (Report No. ARL-TL-6905). US Army Research Laboratory. https://apps.dtic.mil/sti/pdfs/AD1143367.pdf

Chen, J.Y., Lakhmani, S.G., Stowers, K., Selkowitz, A.R., Wright, J.L., & Barnes, M. (2018). Situation awareness-based agent transparency and human-autonomy teaming effectiveness. *Theoretical issues in ergonomics science*, *19*(3), 259-282. https://doi.org/10.1080/1463922X.2017.1315750

Chen, J.Y., Haas, E.C., Pillalamarri, K., & Jacobson, C.N. (2006). *Human-Robot Interface: Issues in Operator Performance, Interface Design, and Technologies* (Report No. ARN-TR-3834). US Army Research Laboratory. https://apps.dtic.mil/sti/pdfs/ADA451379.pdf

Christoffersen, K., & Woods, D.D. (2002). How to make automated systems team players. In E. Salas (Ed.), *Advances in Human Performance and Cognitive Engineering Research* (vol. 2, pp. 1-12). Emerald Publishing Limited. https://doi.org/10.1016/S1479-3601(02)02003-9

Curry, R.E. (1979). Human factors of descent energy management. *1979 18th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, *2*, 422-426. https://doi.org/10.1109/CDC.1979.270209

Dearden, A., Harrison, M., & Wright, P. (2000). Allocation of function: scenarios, context and the economics of effort. *International Journal of Human-Computer Studies*, *52(2)*, 289-318. https://doi.org/10.1006/ijhc.1999.0290

Deci, E.L., & Ryan, R.M. (1987). The Support of Autonomy and the Control of Behavior. *Journal of Personality and Social Psychology*, *53(6)*, 1024-1037. https://doi.org/10.1037/0022-3514.53.6.1024

Defense Science Board Washington DC. (2012). *The Role of Autonomy in DoD Systems* (Task Force Report). US Department of Defense. https://apps.dtic.mil/sti/pdfs/ADA566864.pdf

De Graaf, M.M.A. (2016). An Ethical Evaluation of Human–Robot Relationships. *International Journal of Social Robotics*, *8(4)*, 589-598. https://doi.org/10.1007/s12369-016-0368-5

Dekker, S.W.A. (2002). Reconstructing human contributions to accidents: the new view on error and performance. *Journal of Safety Research*, *33(3)*, 371-385. https://doi.org/10.1016/S0022-4375(02)00032-4

Dekker, S.W.A., & Woods, D.D. (2002). MABA-MABA or Abracadabra? Progress on Human–Automation Co-ordination. *Cognition, Technology & Work*, *4*, 240-244. https://doi.org/10.1007/s101110200022

Demir, M., McNeese, N.J., Johnson, C., Gorman, J. C., Grimm, D., & Cooke, N.J. (2019). Effective Team Interaction for Adaptive Training and Situation Awareness in Human-Autonomy Teaming. In *2019 IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA)* (pp. 122-126). Las Vegas, NV, USA. https://doi.org/10.1109/COGSIMA.2019.8724202

Demir, M., Likens, A.D., Cooke, N.J., Amazeen, P.G., & McNeese, N.J. (2018a). Team coordination and effectiveness in human-autonomy teaming. *IEEE Transactions on Human-Machine Systems*, *49(2)*, 150-159. https://doi.org/10.1109/THMS.2018.2877482

Demir, M., McNeese, N.J., Cooke, N.J. (2018b). Team Synchrony in Human-Autonomy Teaming. In J. Chen (Ed.), *Advances in Human Factors in Robots and Unmanned Systems* (pp. 303-312). Springer. https://doi.org/10.1007/978-3-319-60384-1_29

De Winter, J.C.F., & Dodou, D. (2014). Why the Fitts list has persisted throughout the history of function allocation. *Cognition, Technology & Work*, *16(1)*, 1-11. https://doi.org/10.1007/s10111-011-0188-1

Dubey, A., Kumar, A., Jain, S., Arora, V., & Puttaveerana, A. (2020). HACO: A Framework for Developing Human-AI Teaming. In *Proceedings of the 13th Innovations in Software Engineering*

Conference on Formerly Known as India Software Engineering Conference (pp. 1-9). Association for Computing Machinery. https://doi.org/10.1145/3385032.3385044

Endsley, M.R. (1995). Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors*, *37(1)*, 32-64. https://doi.org/10.1518/001872095779049543

Endsley, M.R., & Kaber, D.B. (1999). Level of automation effects on performance, situation awareness and workload in a dynamic control task. *Ergonomics*, *42(3)*, 462-492. https://doi.org/10.1080/001401399185595

Endsley, M.R., & Kiris, E.O. (1995). The Out-of-the-Loop Performance Problem and Level of Control in Automation. Human Factors, 37(2), 381-394. https://doi.org/10.1518/001872095779064555

Endsley, M.R. (1996). Automation and Situation Awareness. In R. Parasuraman & M. Mouloua (Eds.), *Automation and Human Performance: Theory and Applications* (pp. 163–181). Lawrence Erlbaum Associates, Inc.

Endsley, M.R. (2017). From Here to Autonomy: Lessons Learned from Human–Automation Research. *Human Factors, 59(1)*, 5–27. https://doi.org/10.1177/0018720816681350

Evans, A.W. (2012). *Safe Operations of Unmanned Systems for Reconnaissance in Complex Environments-Army Technology Objective (SOURCE ATO) Field Experimentation Observations and Soldier Feedback* (Report No. ARN-TN-0488). Aberdeen Army Research Laboratory. https://apps.dtic.mil/sti/pdfs/ADA565368.pdf

Feazel, M. (1980). Fuel Pivotal in Trunks' Earnings Slump. *Aviation Week and Space Technology*, *113(18)*, 31-32.

Fiore, S.M., & Wiltshire, T.J. (2016). Technology as Teammate: Examining the Role of External Cognition in Support of Team Cognitive Processes. *Frontiers in Psychology*, *7*, 1531. https://doi.org/10.3389/fpsyg.2016.01531

Fitts, P.M. (1951). *Human Engineering for an Effective Air-Navigation and Traffic-Control System* (Report No. ATI 133 954). Ohio State University. https://apps.dtic.mil/sti/citations/tr/ADB815893

Gao, J., & Dekker, S. (2016). Heroes and Villains in Complex Socio-technical Systems. In A.J. Masys (Eds.), *Disaster Forensics: Understanding Root Cause and Complex Causality* (pp. 47-62). Springer. https://doi.org/10.1007/978-3-319-41849-0_3

Greensteins, J.S., & Lam, S.T. (1985). An Experimental Study of Dialogue-based Communication for Dynamic Human-Computer Task Allocation. *International Journal of Man-Machine Studies*, *23(6)*, 605-621. https://doi.org/10.1016/S0020-7373(85)80061-4

Greenstein, J. S., & Revesman, M. E. (1986). Two Simulation Studies Investigating Means of Human-Computer Communication for Dynamic Task Allocation. *IEEE Transactions on Systems, Man, and Cybernetics*, *16(5)*, 726-730. https://doi.org/10.1109/TSMC.1986.289317

Grimm, D.A., Demir, M., Gorman, J.C., & Cooke, N.J. (2018a). Team Situation Awareness in Human-Autonomy Teaming: A Systems Level Approach. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *62(1)*, 149-149. https://doi.org/10.1177/154193121862103

Grimm, D., Demir, M., Gorman, J.C., & Cooke, N.J. (2018b). The Complex Dynamics of Team Situation Awareness in Human-Autonomy Teaming. In *2018 IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA)* (pp. 103-109). https://doi.org/10.1109/COGSIMA.2018.8423990

Hancock, P.A. (2019). Some Pitfalls in the Promises of Automated and Autonomous Vehicles. *Ergonomics*, *62(4)*, 479-495. https://doi.org/10.1080/00140139.2018.1498136

Hancock, P.A., & Scallen, S.F. (1996). The Future of Function Allocation. *Ergonomics in Design*, *4(4)*, 24-29. https://doi.org/10.1177/106480469600400406

Hollnagel, E. (1991). The Phenotype of Erroneous Actions: Implications for HCI Design. In G. Weir & J. Alty (Eds.), *Human-Computer Interaction and Complex Systems*. London: Academic Press.

Hollnagel, E. (2002). Understanding Accidents - From Root Causes to Performance Variability. In *Proceedings of the IEEE 7th Conference on Human Factors and Power Plants* (pp. 1-1-1-6). https://doi.org/10.1109/HFPP.2002.1042821

Hollnagel, E. (2013). A Tale of Two Safeties. *International Journal of Nuclear Safety and Simulation*, *4(1)*, 1-9. Retrieved from https://www.erikhollnagel.com/A_tale_of_two_safeties.pdf

Hollnagel, E. & Woods, D.D. (2005). *Joint Cognitive Systems: Foundations of Cognitive Systems Engineering*. CRC Press.

Inagaki, T. (2003). Adaptive Automation: Sharing and Trading of Control. In E. Hollnagel (Ed.), *Handbook of Cognitive Task Design* (pp. 147-169). Lawrence Erlbaum.

Janssen, C.P., Donker, S.F., Brumby, D.P., & Kun, A.L. (2019). History and Future of Human-Automation Interaction. *International Journal of Human-Computer Studies*, *131*, 99-107. https://doi.org/10.1016/j.ijhcs.2019.05.006

Johnson, M., Bradshaw, J.M., Feltovich, P.J., Hoffman, R.R., Jonker, C., Riemsdijk, B.V., & Sierhuis, M. (2011a). Beyond Cooperative Robotics: The Central Role of Interdependence in Coactive Design. *IEEE Intelligent Systems*, *26(3)*, 81-88. https://doi.org/10.1109/MIS.2011.47

Johnson, M., Bradshaw, J.M., Feltovich, P.J., Jonker, C., Riemsdijk, B.V., & Sierhuis, M. (2011b). The Fundamental Principle of Coactive Design: Interdependence Must Shape Autonomy. In M. De Vos, N. Fornara, J.V. Pitt, & G. Vouros (Eds.), *Coordination, Organizations, Institutions, and Norms in Agent Systems VI*. Springer. https://doi.org/10.1007/978-3-642-21268-0_10

Johnson, M., Bradshaw, J.M., Feltovich, P.J., Jonker, C., Riemsdijk, B.V., & Sierhuis, M. (2014). Coactive Design: Designing Support for Interdependence in Joint Activity. *Journal of Human-Robot Interaction*, *3(1)*, 43-69. https://doi.org/10.5898/JHRI.3.1.Johnson

Kaber, D.B. & Endsley, M.R. (1997). Out-of-the-Loop Performance Problems and the Use of Intermediate Levels of Automation for Improved Control System Functioning and Safety. *Process Safety Progress*, *16(3)*, 126-131. https://doi.org/10.1002/prs.680160304

Kaber, D.B. (2018). A Conceptual Framework of Autonomous and Automated Agents. *Theoretical Issues in Ergonomics Science*, *19(4)*, 406-430. https://doi.org/10.1080/1463922X.2017.1363314

Kazi, S., Khaleghzadegan, S., Dinh, J.V., Shelhamer, M.J., Sapirstein, A., Goeddel, L.A., Chime, N.O., Salas, E., & Rosen, M.A. (2019). Team Physiological Dynamics: A Critical Review. *Human Factors*, 63(1), 32–65. https://doi.org/10.1177/0018720819874160

Klein, G., Woods, D.D., Bradshaw, J.M., Hoffman, R.R., & Feltovich, P.J. (2004). Ten Challenges for Making Automation a "Team Player" in Joint Human-Agent Activity. *IEEE Intelligent Systems*, *19*(6), 91-95. https://doi.org/10.1109/MIS.2004.74

Le Bars, S., Devaux, A., Nevidal, T., Chambon, V., & Pacherie, E. (2020). Agents' Pivotality and Reward Fairness Modulate Sense of Agency in Cooperative Joint Action. *Cognition*, *195*, 104117. https://doi.org/10.1016/j.cognition.2019.104117

Lee, J.D., & See, K.A. (2004). Trust in Automation: Designing for Appropriate Reliance. *Human Factors*, *46(1)*, 50-80. https://doi.org/10.1518/hfes.46.1.50_30392

Le Goff, K., Rey, A., Haggard, P., Oullier, O., & Berberian, B. (2018). Agency Modulates Interactions with Automation Technologies. *Ergonomics*, *61(9)*, 1282-1297. https://doi.org/10.1080/00140139.2018.1468493

Lin, P. (2016). Why Ethics Matters for Autonomous Cars. In M. Maurer, J. Gerdes, B. Lenz, & H. Winner (Eds.), *Autonomous Driving* (pp. 69-85). Springer. https://doi.org/10.1007/978-3-662-48847-8_4

Lyons, J.B. (2013). Being Transparent About Transparency: A Model for Human-Robot Interaction. In *Papers from the 2013 AAAI Spring Symposium*.

Lyons, J.B., Sadler, G.G., Koltai, K., Battiste, H., Ho, N.T., Hoffmann, L.C., Smith, D., Johnson, W., & Shively, R. (2017). Shaping Trust Through Transparent Design: Theoretical and Experimental Guidelines. In P. Savage-Knepshield & J. Chen (Eds.), *Advances in Human Factors in Robots and Unmanned Systems* (pp. 127-136). Springer. https://doi.org/10.1007/978-3-319-41959-6_11

McCafferty , D.B. , Hendrikse , E.J., & Miller , G.E. (2004). Human Factors Engineering (HFE) and Cultural Calibration for Vessel and Offshore Installation Design. In M. Kaplan (Ed.), *Cultural Ergonomics: Volume 4* (pp. 105-145). Emerald Group Publishing Limited. https://doi.org/10.1016/S1479-3601(03)04004-9

Mercado, J.E., Rupp, M.A., Chen, J.Y., Barnes, M.J., Barber, D., & Procci, K. (2016). Intelligent Agent Transparency in Human–Agent Teaming for Multi-UxV Management. *Human Factors*, *58(3)*, 401-415. https://doi.org/10.1177/0018720815621206

Mertes, F., & Jenney, L. (1974). *Automation Applications in an Advanced Air Traffic Management System: Volume 3. Methodology for Man-Machine Task Allocation* (Report No. DOT-TSC-OST-74-14-3). United States Department of Transportation. https://rosap.ntl.bts.gov/view/dot/11617/dot_11617_DS1.pdf

Meshkati, N., & Khashe, Y. (2015). Operators' Improvisation in Complex Technological Systems: Successfully Tackling Ambiguity, Enhancing Resiliency and the Last Resort to Averting Disaster. *Journal of Contingencies and Crisis Management*, *23(2)*, 90-96. https://doi.org/10.1111/1468-5973.12078

Miller, M.J., & Feigh, K.M. (2019). Addressing the Envisioned World Problem: A Case Study in Human Spaceflight Operations. *Design Science*, *5*, E3. https://doi.org/10.1017/dsj.2019.2

Moray, N. (1986). Monitoring Behavior and Supervisory Control. In K.R. Boff, L. Kaufman, & J.P. Thomas (Eds.), *Handbook of Perception and Human Performance, Vol. 2. Cognitive processes and performance* (pp. 1–51). John Wiley & Sons.

Mouloua, M., Parasuraman, R., & Molloy, R. (1993). Monitoring Automation Failures: Effects of Single and Multi-Adaptive Function Allocation. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *37(1)*, 1-5. https://doi.org/10.1177/1541931293037001

Norman, D.A. (1993). *Things That Make Us Smart: Defending Human Attributes in the Age of the Machine*. Basic Books.

Older, M.T., Waterson, P.E., & Clegg, C.W. (1997). A Critical Assessment of Task Allocation Methods and Their Applicability. *Ergonomics*, *40(2)*, 151-171. https://doi.org/10.1080/001401397188279

Pagliari, M., Chambon, V., & Berberian, B. (2022). What is New with Artificial Intelligence? Human–Agent Interactions Through the Lens of Social Agency. *Frontiers in Psychology*, *13*, 954444. https://doi.org/10.3389/fpsyg.2022.954444

Parasuraman, R., Molloy, R., & Singh, I.L. (1993). Performance Consequences of Automation-Induced "Complacency". *The International Journal of Aviation Psychology*, *3(1)*, 1–23. https://doi.org/10.1207/s15327108ijap0301_1

Parasuraman, R., Sheridan, T.B., & Wickens, C.D. (2000). A Model for Types and Levels of Human Interaction with Automation. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, *30(3)*, 286-297. https://doi.org/10.1109/3468.844354

Parasuraman, R., Sheridan, T.B. and Wickens, C.D., (2008). Situation Awareness, Mental Workload, and Trust in Automation: Viable, Empirically Supported Cognitive Engineering Constructs. *Journal of cognitive engineering and decision making*, *2(2)*, 140-160. https://doi.org/10.1518/155534308X284417

Pernice, K. (2018) User interviews 101. Retrieved from:. https://www.nngroup.com/articles/user-interviews/

Portigal, S. (2013). *Interviewing users.* Rosenfeld media

Price, H.E. (1985). The Allocation of Functions in Systems. *Human Factors*, *27(1)*, 33-45. https://doi.org/10.1177/001872088502700104

Reason, J. (2008). The Human Contribution: Unsafe Acts, Accidents, and Heroic Recoveries. CRC Press. https://doi.org/10.1201/9781315239125

Rouse, W.B. (1981). Human-Computer Interaction in the Control of Dynamic Systems. *ACM Computing Surveys*, *13*(1), 71-99. https://doi.org/10.1145/356835.356839

Rouse, W.B. (1988). Adaptive Aiding for Human/Computer Control. *Human Factors*, *30*(4), 431-443. https://doi.org/10.1177/001872088803000405

Rouse, W.B. (1994). Twenty Years of Adaptive Aiding: Origins of the Concepts and Lessons Learned. In M. Mouloua, & R. Parasuraman (Eds.), Human Performance in Automated Systems: Current Research and Trends (pp. 28-33). Lawrence Erlbaum.

Russell, S., & Norvig, P. (2014). *Artificial Intelligence: A Modern Approach (3rd ed.)*. Pearson.

Salas, E., Sims, D.E., & Burke, C.S. (2005). Is There a "Big Five" in Teamwork? *Small Group Research*, *36(5)*, 555-599. https://doi.org/10.1177/1046496405277134

Salmon, P., Jenkins, D., Stanton, N., & Walker, G. (2010). Hierarchical Task Analysis vs. Cognitive Work Analysis: Comparison of Theory, Methodology and Contribution to System Design. *Theoretical Issues in Ergonomics Science*, *11(6)*, 504-531. https://doi.org/10.1080/14639220903165169

Sanders, T.L., Wixon, T., Schafer, K.E., Chen, J.Y.C., & and Hancock, P.A. (2014). The Influence of Modality and Transparency on Trust in Human-Robot Interaction. In *2014 IEEE International Inter-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)* (pp. 156-159). https://doi.org/10.1109/CogSIMA.2014.6816556

Sarter N.B., Woods D.D., & Billings C.E. (1997). Automation Surprises. In G. Salvendy (Ed.), *Handbook of Human Factors and Ergonomics* (pp. 1926–1943). Wiley.

Scerbo, M. (2007). Adaptive Automation. In R. Parasuraman, & M. Rizzo (Eds.), *Neuroergonomics: The Brain at Work* (pp. 239-252). Oxford University Press.

Sebanz, N., Knoblich, G., & Prinz, W. (2003). Representing Others' Actions: Just Like One's Own? *Cognition*, *88(3)*, B11-B21. https://doi.org/10.1016/S0010-0277(03)00043-X

Sebanz, N., Knoblich, G., & Prinz, W. (2005). How Two Share a Task: Corepresenting Stimulus–Response Mappings. *Journal of Experimental Psychology: Human Perception and Performance*, *31(6)*, 1234–1246. https://psycnet.apa.org/doi/10.1037/0096-1523.31.6.1234

Selkowitz, A.R., Lakhmani, S.G., Larios, C.N., & Chen, J.Y. (2016). Agent Transparency and the Autonomous Squad Member. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (pp. 1319-1323). https://doi.org/10.1177/1541931213601613

Shannon, C.J., Horney, D.C., Jackson, K.F., & How, J.P. (2017). Human-Autonomy Teaming Using Flexible Human Performance Models: An Initial Pilot Study. In *Advances in Human Factors in Robots and Unmanned Systems* (pp. 211-224). Springer. https://doi.org/10.1007/978-3-319-41959-6_18

Sheridan, T.B. (2011). Adaptive Automation, Level of Automation, Allocation Authority, Supervisory Control, and Adaptive Control: Distinctions and Modes of Adaptation. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, *41(4)*, 662-667. https://doi.org/10.1109/TSMCA.2010.2093888

Sheridan, T.B. (1997). Task Analysis, Task Allocation and Supervisory Control. In M.G. Helander, T.K. Landauer, & P.V. Prabhu (Eds.), *Handbook of Human-Computer Interaction* (2nd ed., pp. 87-105). Elsevier. https://doi.org/10.1016/B978-044481862-1.50071-6

Sheridan, T.B., & Verplank, W.L. (1978). *Human and Computer Control of Undersea Teleoperators*. Man-Machine Systems Laboratory, Department of Mechanical Engineering, MIT. https://apps.dtic.mil/sti/pdfs/ADA057655.pdf

Smith, C. (2020). Designing Trustworthy AI: A Human-Machine Teaming Framework to Guide Development. *Carnegie Mellon University*. https://doi.org/10.1184/R1/12119847.v1

Stanton, N.A. (2006). Hierarchical Task Analysis: Developments, Applications, and Extensions. *Applied Ergonomics*, *37*, 55-79. https://doi.org/10.1016/j.apergo.2005.06.003

Stowers, K., Kasdaglis, N., Newton, O., Lakhmani, S., Wohleber, R., & Chen, J. (2016). Intelligent Agent Transparency: The Design and Evaluation of an Interface to Facilitate Human and Intelligent Agent Collaboration. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *60(1)*, 1706-1710. https://doi.org/10.1177/1541931213601392

Stubbs, K., Hinds, P.J., & Wettergreen, D. (2007). Autonomy and Common Ground in Human-Robot Interaction: A Field Study. *IEEE Intelligent Systems*, *22(2)*, 42-50. https://doi.org/10.1109/MIS.2007.21

Swain, A.D., & Guttman, H.E. (1983). *Handbook of Human-Reliability Analysis with Emphasis on Nuclear Power Plant Applications. Final Report* (Report No. NUREG/CR-1278; SAND-80-0200). Sandia National Lab (SNL-NM), Albuquerque, NM. https://doi.org/10.2172/5752058

Sycara, K., & Sukthankar, G. (2006). Literature Review of Teamwork Models. *Robotics Institute, Carnegie Mellon University*, *31*(31), 1-31.

Trotter, M.J., Salmon, P.M., & Lenné, M.G. (2013). Improvisation: Theory, Measures and Known Influencing Factors. *Theoretical Issues in Ergonomics Science*, *14(5)*, 475-498. https://doi.org/10.1080/1463922X.2012.656153

Vanderelst, D., & Winfield, A. (2018). An Architecture for Ethical Robots Inspired by the Simulation Theory of Cognition. *Cognitive Systems Research*, *48*, 56-66. https://doi.org/10.1016/j.cogsys.2017.04.002

Van der Wel, R.P.R.D. (2015). Me and We: Metacognition and Performance Evaluation of Joint Actions. *Cognition*, *140*, 49-59. https://doi.org/10.1016/j.cognition.2015.03.011

Van der Wel, R.P.R.D., Sebanz, N., & Knoblich, G. (2012). The Sense of Agency During Skill Learning in Individuals and Dyads. *Consciousness and Cognition*, *21*(3), 1267-1279. https://doi.org/10.1016/j.concog.2012.04.001

Wegner, D.M., Sparrow, B., & Winerman, L. (2004). Vicarious Agency: Experiencing Control Over the Movements of Others. *Journal of Personality and Social Psychology*, *86*(6), 838-848. https://doi.org/10.1037/0022-3514.86.6.838

Wen, W., & Imamizu, H. (2022). The Sense of Agency in Perception, Behaviour and Human-Machine Interactions. *Nature Reviews Psychology*, 1, 211–222. https://doi.org/10.1038/s44159-022-00030-6

Wickens, C.D., Hollands, J.G., Banbury, S., & Parasuraman, R. (2013). *Engineering Psychology and Human Performance (4th ed.)*. Pearson.

Wiener, E.L. (1988). Cockpit Automation. In E.L. Wiener & D.C. Nagel (Eds.), *Human Factors in Aviation* (pp. 433–461). Academic Press. https://doi.org/10.1016/B978-0-08-057090-7.50019-9

Wilson, C., (2013). *Interview techniques for UX practitioners: A user-centred design method.* Morgan Kaufmann

Wohleber, R.W., Stowers, K., Chen, J.Y., & Barnes, M. (2017). Effects of Agent Transparency and Communication Framing on Human-Agent Teaming. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 3427-3432). IEEE. https://doi.org/10.1109/SMC.2017.8123160

Woods, D.D. (1985). Cognitive Technologies: The Design of Joint Human-Machine Cognitive Systems. *AI Magazine*, *6(4)*, 86-92. https://doi.org/10.1609/aimag.v6i4.511

Woods, D.D. & Dekker, S.W.A. (2000). Anticipating the Effects of Technological Change: A New Era of Dynamics for Human Factors. *Theoretical Issues in Ergonomic Science*, *1(3)*, 272–282. https://doi.org/10.1080/146392201100374 52

Appendix A Conducting interviews

Interviews can be held to gather information about the context where a system is used, user needs or as a part of an evaluation of systems or functions, among other things. Interviewing is a skill that takes practice and this guide will give you some useful advice on how to do it.

This guide primarily focuses on a semi-structured interview which allows the interviewer to combine predefined topics and questions with a freer flowing interview style. This interview technique will allow the interviewer gather systematic information in order to be able compare the answers of the respondents on the same set of questions, while allowing the flexibility to pursue new topics as needed. The section below describes tasks/steps to consider *before*, *during* and *after* conducting an interview in order to accomplish a successful interview.

A.1 Before

Phases of the interview process	Examples and tips
Prepare In the preparation work before an interview, you define a goal with the interview, decide on who to interview, how to collect the data, prepare questions and inform participants of what to expect.	**Materials that can be helpful to prepare:** Interview project plan describing goals, recruiting plan, background on companies to visit, general topics of interest, data collection and analysis plan. Even small projects can benefit from a project plan. • A letter of introduction to send to participants. • Informed consent forms that explain the purpose of the study, how the data will be used, and permission for data recordings. • Interview guides

Define a goal with the interview:
 	
Setting goals for the interview will help select questions and a direction for the interview. Depending on if you are conducting the interview for the system model, allocation model, implementation phase or for evaluation purposes, the goal is different. The goal when performing an interview can of course vary widely but could potentially be one or more of the following:
 	
• Understand the respondents perspective with regards to the topic or problem.
• Deepen the knowledge of a task, such as:
  ○ The steps behind the task.
  ○ Agents (such as organizations, human actors/operators and artificial agents and systems) that are associated to the task and how do they behave.
  ○ Tools that are involved and what information is exchanged.
  ○ Cognitive issues are connected to a certain task.
	Examples of interview goals:
	
	How do nurses feel about logging medical data, and what are the processes they believe they use?
	
	Learn how architects share CAD drawings with engineers, and where they feel there are challenges and opportunities.
	
	Find out how bicycle couriers get the best route directions, and what they feel works well, where they think there are issues, and how they think things could be improved.

Phases of the interview process	Examples and tips
o Map workflows and try to understand what is efficient/inefficient. • Understand the context in which de respondent operates, such as: o The roles and groups that exists and what their goals and tasks are. o Interaction between roles and groups. o Systems used by other roles and groups. o Physical work environments and workplace cultures. • Identify critical activities and situations o Event analysis (critical events) o Domain critical things, as safety • Testing ideas or hypotheses from other sources	
Decide who should participate and how to collect data Plan how the data will be collected during the interview. Taking notes and recording the interview are two methods that work well together. With good notetaking during the interview, the recording can be used only when clarification is needed and the time-consuming process to transcribe might be possible to avoid. It is important to consider the privacy or the respondent and have a plan for how you want to handle the data after the interview. When taking notes, you can avoid writing down any sensitive and personal information the respondent might share, but with a recording you must plan how to handle such data. For instance, how the recording will be stored. Pay attention to what information must be dealt with according to GDPR. Decide which team members should participate in the interview. It might be beneficial to conduct the interview in a team of two. In such a team one person can take notes and allow the other person to focus on the dialogue with the respondent. The person who is focusing on the notes will likely also have some follow-up questions that they want to ask during the interview, and you can decide beforehand how to handle that.	**Interview environment and artifacts** Conducting an interview on site provides great opportunity to form an understanding of the culture. Each site is filled with artifacts that can help you understand users and their environment.
Prepare a set of questions and discussion topics The interview can be *structured*, *semi-structured* or *unstructured*. In the structured interview, the questions are fixed according to a script. In the unstructured only the topic of the interview is fixed, and the semi-structured is a mix of the two. Unstructured interviews are usually more challenging (Wilson, 2013). This section will focus on semi-structured interviews. The semi-structured interview is useful for understanding user goals and to gather information about tasks, task flow, and work artifacts. A structured interview on the other hand, is mentioned to be useful for instance when results are to be compared across different group of users (Wilson, 2013). For the semi-structured interview, it is good to prepare an interview guide containing questions or topics you want to have answered.	**Example of questions suitable to apply in many interviews** • What is your current role in and your organization? o And before that? • What is a description of a typical day/week/month at your job? • How do you use a product/function/service? • What are the problems with this product/function/process/service? • What are the best things about…?

Phases of the interview process	Examples and tips
The interview guide will help you to keep the interview on track, ensure that you do not miss anything and might help you relax as an interviewer. It may also help you ask clear non-leading questions and is a way to get you team's input on questions and topics before the interview (Pernice, 2018). Wilson (2013) lists that the interview guide typically consists of: • An introduction to the purpose and topic of the interview • A list of topics and questions to ask about each topic • Suggested probes and prompts • Closing comments. Avoid questions that are too long or complex and use probing questions to get more detailed picture of the subject discussed. Also avoid double questions, break them into two separate questions. Along with the prepared questions, it is often helpful to include a list of neutral prompts in the interview guide in order to avoid starting questions with leading prompts. (Wilson, 2013). In the semi-structured interview, you don't have to ask the questions exactly as they are phrased in the guide.	• What tools, software, or hardware do you use to accomplish your goals? How often do you use them? • Can you give me a specific example? **Neutral prompts to use (Wilson, 2013):** • "Tell me about. . ." • "Could you explain a bit more what you meant by. . .?" • "How do you feel about. . .?" • "Could you describe. . .?" **Topic example** In the third phase of the SafeTeam Framework a set of questions are presented intended to evaluate the newly designed systems. Inspiration can be drawn from this list to create topics to identify flaws with existing systems as well.

Inform respondent about the process

Prepare information for the participant before the interview. This can inform the respondent about the duration of the interview, the purpose of the interview and how the data will be *used* and *stored*. Also inform the participant if the interview is anonymous or not. This information can be included in an *Informed Consent* form.

A.2 During

Phases of the interview process	Examples and tips
Getting started Before the interview begins, there might be some set-up needed e.g., recording device. Make sure to interfere as little as possible with the respondent's space if you are visiting their environment, thus be mindful when setting up any equipment (Wilson, 2013). Ideally, the respondent is already informed on the interview process, but make sure to refresh their memory about purpose, topic, how data will be collected and used and interview time. As you move on to the actual interview, it is a good practice to warm-up the participant with some introductory questions that should be perceived as easy, nonthreatening, and relevant. This might be a good time to collect some background information about the participant and understand the context in which they operate (Wilson, 2013).	

Phases of the interview process	Examples and tips
Interview recommendations During the main part of the interview, it is a good habit to start off with the questions that you want every respondent to answer, and then ask the remaining questions. The questions are modified as needed during the interview, depending on the answers from the respondents (Wilson, 2013). As questions are modified and follow-up questions are asked, make use of the list of neutral prompts. Remember to follow up with asking 'why?'. In the later analysis, you might find contradicting opinions or opinions that you wonder should result in a design decision or not. If you know the reason for the opinion, the decision might be easier to make. You should give the respondent time to finish their thoughts and make sure to not interrupt them. You should also not be afraid of silence. You will often find that the respondent will continue talk if they notice that you do not move on to the next question. Portigal (2013) explains this as people talking in paragraphs and wanting permission to move on to the next paragraph, which is what you give them when you do not move on to the next question. Naturally it is important that the participant feels comfortable. However, keep in mind that there's a big difference between *rapport* and *friendship*. The user does not have to really like you, think you're funny, or want to invite you out for a cup of coffee in order to trust you enough to be interviewed (Pernice, 2018).	**Follow-up with concrete examples** It can sometimes be easier or more natural for the participant to talk about specific events than general processes (Pernice, 2018). The general process might be that 'the task is always carried out successfully'. When you ask the participant to walk you through the last time they did it, you might on the other hand find out that they have a specific workaround for when it rains or that a particular task is more cumbersome and that they must be two persons to carry it out.

Wrapping up

Make sure you indicate clearly when the interview is over by putting away note-taking materials and turn off any potential recording devices, make sure to thank the respondent for participating in the interview. Ask if it is okay to contact the participant if any further questions may arise during data analysis and interpretation (Wilson, 2013).

A.3 After

Phases of the interview process	Examples and tips
Analysis Before starting with the analysis, preparations might be necessary. A couple of things might have to be done immediately while other things, such as transcribing if you choose to that, can be done later. As mentioned previously, transcribing is very time-consuming and you may choose to rely on your notes and recordings.	
As soon as possible Consider scheduling time for a debriefing with the interviewing team immediately after each session. Portigal (2013) points out that already the next day your memory starts to fade and suggests going for food and talk after the interview. But make sure to make notes! The time shortly after an interview is also a good time to clean up your notes and clarify things that there was not time to note at the time, if necessary.	

Phases of the interview process	Examples and tips
Analyze collected data There are many ways to go through your material and analyze the data. One way is to collect quotes or insights from all interviews and group in different themes. You can start with high level themes and later return to each theme and sort further under sub-themes. Remember to keep a reference to the source with each quote. This serves two purposes: Making it easier to go back to that specific interview if you (or a colleague) want more background to the quote. Ensuring that you do not later treat one person's three related opinions as three users' opinions. Other ways of analyzing are presented to the right.	**Inductive Methods of Analyzing Interview Transcripts** *A thematic content analysis* begins with weeding out biases and establishing your overarching impressions of the data. Rather than approaching your data with a predetermined framework, identify common themes as you search the materials organically. Your goal is to find common patterns across the data set. The goal of thematic content analysis is to find common patterns across the data set. A *narrative analysis* involves making sense of your interview respondents' individual stories. Use this type of qualitative data analysis to highlight important aspects of their stories that will best resonate with your readers. And, highlight critical points you have found in other areas of your research. **Deductive Approach to Qualitative Analysis** *Deductive analysis*, on the other hand, requires a structured or predetermined approach. In this case, the researcher will build categories in advance of their analysis. Then, they'll map connections in the data to those specific categories.