REGULAR ARTICLE

WILEY

PRODEC for human systems integration of increasingly autonomous systems

Guy André Boy[1,2] 🔘 | **Dimitri Masson**[2] | **Élise Durnerin**[2] | **Chloé Morel**[2]

[1]FlexTech Chair, CentraleSupélec, Paris Saclay University, Gif-sur-Yvette, France

[2]FlexTech Chair, ESTIA Institute of Technology, Bidart, France

Correspondence
Guy André Boy, FlexTech Chair, CentraleSupélec, Paris Saclay University, 9 rue Joliot Curie, 91190 Gif-sur-Yvette, France.
Email: guy-andre.boy@centralesupelec.fr

Abstract

The purpose of the PRODEC scenario-based design method is the incremental cross-fertilization and refinement of procedural scenarios and declarative configurations. It uses virtual prototypes of the developed systems (mainly life-critical systems) to conduct human-in-the-loop simulations (HITLSs). Based on human systems integration (HSI) principles and criteria, as well as expertise and experience in the domain at stake, this HSI approach grounded in virtual environments requires a clear definition of physical and cognitive tangibility metrics to assess the distance between virtual and tangible (grasp) dimensions of the system being developed when it is put to work. PRODEC considers human and machine systems described in terms of structures and functions incrementally designed using procedural scenarios (i.e., stories) in task and activity networks, which provide life to declarative configurations (i.e., system's functions and structures). This active modeling and simulation process enables the discovery of emergent structures and functions of the system being developed when it is virtually operated. PRODEC's use is illustrated in an example. We discuss the use of PRODEC and its results as to how they can be used with digital twins.

KEYWORDS
BPMN, complexity, digital engineering, flexibility, human systems integration, human-centered design, human-in-the-knowledge elicitation, maturity, modeling, oop simulation, physical and cognitive structure and functions, procedural and declarative scenarios, stability, sustainability, tangibility, virtual prototypes

1 | INTRODUCTION

First, we must distinguish between traditional human factors and ergonomics (HF&E) evaluation approaches and human systems integration (HSI) design approaches. HSI is not an add-on technique to adapt people to engineering systems already designed and developed (i.e., the user interface syndrome, a relic of the 20th century). Nowadays, "HSI is an essential, transdisciplinary, sociotechnical, and management approach of systems engineering to ensure that the system's technical, organizational, and human elements are appropriately addressed across the whole system life cycle, service, or enterprise system. HSI considers systems in their operational context and the necessary interactions between and among their human and technological elements to make them work in harmony and cost-effectively, from concept to retirement".[1] HSI intersects several fields, mainly systems engineering (SE), HF&E, and information technology, not to forget the expertise field (e.g., health, defense, transportation).

In addition, some engineers may say that what we do in the field of HSI is not as "sexy" as what is done in basic SE. This is because we need to be aware, in the sense of a deep understanding of what

people do in their work and activities. To put it more formally, instead of simplifying a complex socio-technical system (STS) and applying a "sexy" SE problem-solving method, we put our energy into familiarizing ourselves with the complexity of that system. To do this, we need to get close to the people in the field, interview them, observe what they do, listen to their complaints, etc. This takes time and effort, just as anthropologists do when they try to familiarize themselves with the activities of a tribe. It is worth noting that simplification always leads to a simplistic solution that will likely end up with surprises; on the contrary, familiarizing yourself with a complex STS highlights tangible elements that can be studied, rethought, and improved. HSI proposes methods and tools for this familiarization process and further human-centered design processes. This article suggests clarifying this necessary evolution by presenting the PRODEC method. PRODEC is an HSI method that combines the HF&E and SE fields and is applied in a case study involving replacing human operators with robots for routine operations and maintenance of a remote oil and gas platform. In such an example, PRODEC uses an incrementally fine-tuned digital twin of the platform to iteratively understand and improve design, development, and operations of the complex systems at stake.

PRODEC enables the integration of activity analysis at the design stage. **Activity** is the effective result of executing a prescribed **task** initially prescribed in an STS. For example, a task could be "if the pressure in pipe 8 is bigger than 50 psia then check the temperature of the tank 12, and <a series of checks and actions>," as observed activity of an expert human operator is "if the pressure in the lower part of the installation is high then turn valve 7 to potentially stop fuel leak." Activity results from various context considerations, unexpected disturbances, and/or the expertise of experienced human operators. Only recently, activity analysis was carried out on an existing system before a new one was planned to replace it, but not during development. Why? This is because activity analysis can provide useful and usable results when the system is fully integrated, not tested part by part. Digital engineering makes this possible by using virtual prototypes to create and operate human-in-the-loop simulations (HITLS). PRODEC is a scenario-based design (SBD) method[2] augmented by HITLS to enable activity observation and analysis and incremental discovery and integration of emergent system properties during design and development throughout the system life cycle. Emergence typically occurs when unexpected situations happen (i.e., what is not anticipated in regular procedures and automation). This follows the "learning by using" approach.[3](pp. 120–140)

Another dimension of activity is joint activity,[4] where teammates often back each other up or support each other during certain activities. This is not necessarily anticipated; function allocation must be dynamic in real-time. However, when procedures and conventional automation cannot support this kind of thing, it must be supported using what we call "FlexTech" (i.e., technology for flexibility). PRODEC is unique for discovering emergent properties in the form of functions and structures that support FlexTech design. An example is the "undo" function available in text editing. Without it, writing is a highly rigid task and becomes flexible with it.

Within digital engineering, we decided to focus PRODEC on the engineering design of **life-critical sociotechnical systems of systems (SoSs) that are becoming increasingly autonomous**. For example, we used PRODEC to investigate the collaboration of fighter pilots and virtual assistants onboard air combat jets, the remote management of a fleet of mobile robots on offshore oil and gas platforms, and a national health system. In these studies, virtual prototypes are typically developed to support HITLS and permit observing human and machine activities.

However, if digital engineering can support serious consideration of human factors studies at the design stage, it is based on a digital account of the real system to be developed. This is why physical tangibility and figurative (or cognitive) tangibility have become crucial concepts that will be further defined in this article. A physical object or an abstract entity is tangible when we can physically or figuratively grasp it. Either you can touch, feel, or understand it. The concept of **tangibility** has already been described along five factors[5]: complexity, maturity, flexibility, stability, and sustainability. Since PRODEC starts in a virtual world, solutions are created and incrementally modified, considering the results of formative evaluations based on a concurrent association of the chosen tangibility factors. We distinguish between formative evaluations, which enable us to provide recommendations for design improvements, and summative evaluations, which enable us to certify the final product.

PRODEC also addresses the crucial issue of unexpected situations, especially in socio-technical systems where human life is at stake, that is, in life-critical systems (LCSs). For a long time, LCSs have been based on principles, methods, and tools that support optimizing safety, efficiency, and comfort. This is why technology-centered approaches led to operational procedures and automation development and use. Procedures automate people, and technological automation results from implementing procedures into machines, therefore automating machines. In both cases, human–machine LCSs are performing in rigid procedures and automation based on expected situations. However, LCS designs require different principles, methods, and tools to maintain system resilience when unexpected events occur. They require operational flexibility. Therefore, it is urgent to have modeling and simulation support as early as possible during the design and development processes to observe activity resulting from task execution and discover emergent behaviors and properties that induce design changes in the form of appropriate structures and functions, ensuring safety, efficiency, and comfort in any situation, including unexpected[*] ones. This process requires a reasonable amount of time to ensure the maturity of technology, induced practice, and organizational issues. We typically start from low-fidelity prototypes that are incrementally upgraded through a series of formative evaluations toward increasing maturity.[6] Indeed, it requires time and reflection mixed with trial and error, so HITLS must be available to operators for substantial periods. This is the

---

[*] Note that the term "unexpected" includes known and unknown situations. HITLS enables the discovery of several situations unexpected by engineering design teams, but not all. "Unknown" situations can only be discovered at any time during the life cycle of the sociotechnical system at stake. Therefore, we cannot implement the "unexpected" concept without considering experience feedback processes seriously.

price to pay for avoiding endless corrections and considerable expense at the end of production, delivery, and, ultimately, operations cycles, just because HSI was not adequately considered at design time.

Consequently, PRODEC has been developed to capture and combine procedural knowledge from operations people and declarative knowledge used by engineering design and systems engineering people. Procedural knowledge covers normal, abnormal, and emergency operations in expected and unexpected situations. It involves humans and machines. Declarative knowledge covers the design of structures and functions of the various agents and objects involved in a large set of scenarios and configurations. In summary, automating people and machines in expected situations defines "rigid automation," and supporting people's problem-solving in unexpected situations defines "flexible autonomy." Automation is made of predefined procedures that are executed as a reaction to an event or a situation. Autonomy provides learning and adaptation properties to a system in dynamic environments and, therefore, evolves from successful and unsuccessful interactions with these environments.

From this point of view, PRODEC differs from classical task analysis methods that supports the design of user interfaces and, more importantly, emphasizes what is prescribed, hoping that people and machines will apply procedures precisely as defined.[7,8] PRODEC process starts by acquiring knowledge from field experience to make appropriate scenarios from operations, engineering design, and a harmonious combination of both. Consequently, a rational approach dedicated to knowledge acquisition and knowledge engineering[9–12] has been developed.

Unlike most engineering design methods,[13,14] PRODEC associates creativity with systems engineering and operational experience during the whole life cycle of the system at stake, more specifically from the very beginning of its design. Experience is typically based on existing good practice and therefore fostering conservative behaviors, as creativity and innovation result in changes creating and developing new concepts, therefore fostering out-of-the-box behaviors. It is usual that "people resist change."[15] We claim that creativity and experience should always be combined when we deal with complex STS design and development to anchor innovative solutions into field reality.

"The hardest part of design is getting the requirements right, which means ensuring that the right problem is solved and the solution is appropriate. Requirements made in the abstract are invariably wrong. Requirements produced by asking people what they need are invariably wrong. Requirements are developed by watching people in their natural environment."[16] Indeed, watching people in their natural environment enables engineering designers and systems engineers to understand the complexity of human operators' work.

PRODEC, as an activity-centered design approach, requires another dimension: identification of incremental emergence of system behaviors and properties. This article expands on what we have already illustrated in the MOHICAN project on the human-centered integration of a virtual assistant system in a combat aircraft.[17] More generally, the development of cognitive systems (i.e., systems incorporating artificial intelligence [AI] algorithms) that provide new trust and collaboration issues requires a more structured approach to HSI,[18,19]
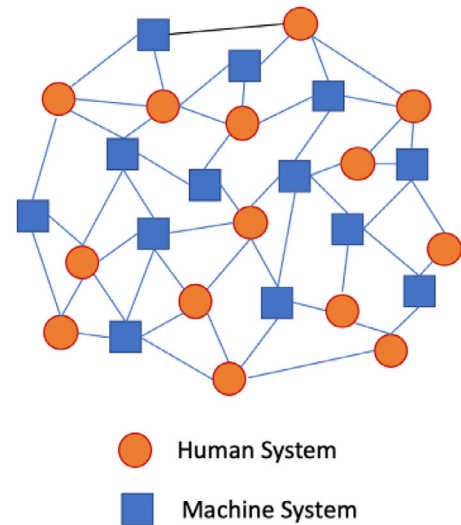

FIGURE 1 Human and machine systems interacting with each other.
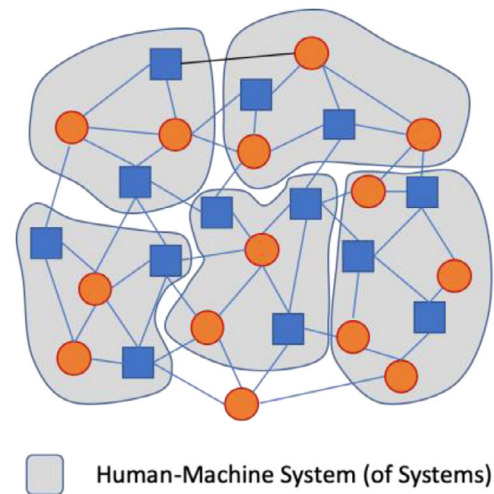

FIGURE 2 Systems of systems interacting with each other.

understood as a combination of systems engineering (SE) and virtual human-centered design (VHCD).[20–22] Therefore, we claim that this approach must address the integration of humans and cognitive systems in many environments and be compatible with multi-agent methods developed in AI, where an agent is considered a society of agents.[23] AI and SE have much in common,[24] and specifically, an agent in AI can be considered a system in SE. More recent human-agent teaming models have been developed considering organization, role, competency, and task.[25]

Consequently, HSI must consider a common system representation of humans and machines. This leads to multi-agent representations, such as in Figure 1, and a society of agents can be considered an SoS (Figure 2). Systems have structures and functions that must be identified. They can be cognitive and/or physical.[26] In addition, a large body of work is available on the use of expert systems[27–30] and cognitive

ergonomics.[31] Consequently, this background will be systematically used in PRODEC to support systems engineers.[32–35]

PRODEC is based on a **robust systems representation framework** that supports the description of organizational aspects of human–machine systems of systems.[26,36] Previous contributions to HSI on modeling approaches have mainly focused on extensions of the System Modeling Language (SysML) (Madni and Madni[37]; Madni and Orellana[38]; Orellana and Madni[39]; Raymond and Prun[40]; Watson et al.[41]). These approaches do not ensure the coherence and interoperability of human and mechanical system representations. Consequently, their conceptual foundations need to be more mature, and the conceptual cross-fertilization with human-centered design (HCD) needs to be clarified. Therefore, there is still a need to develop further and refine conceptual models for HSI using previously developed concepts.[17]

To summarize, PRODEC intersects several fields, including knowledge management (KM), HF&E, human–computer interaction (HCI), and SE. Early KM attempts focused on developing declarative knowledge representations rather than SBD.[42,43] Conversely, HF&E deeply developed methods based on activity analysis but less on computer-tractable knowledge representation.[44–46] The HCI community started the human-centered design (HCD)[†] approach but typically not for complex socio-technical systems.[16,48] Within the evolution of SE and recent development of HSI, four problems need to be solved: (1) how procedural knowledge elicited from operations people and declarative knowledge elicited from engineering designers and systems engineers could be combined effectively (i.e., the systemic aspects and the contextual aspects of an STS); (2) how the antagonism between experience and creativity can be overcome; (3) how could we create an environment where the activity (i.e., the way tasks are executed) could be observed and further analyzed, significantly augmenting traditional task analysis; (4) what kind of robust systems representation framework could support the description of organizational aspects of multi-agent systems of systems. We propose to present the PRODEC formal approach to HSI, which proposes solutions to these four problems. We will use a real-world case study to show how PRODEC can be used concretely. A discussion and comparison to other research efforts will be provided. Finally, we will conclude and give perspectives on PRODEC.

2 | PRODEC: A FORMAL APPROACH TO HUMAN SYSTEMS INTEGRATION

Since the PRODEC methodology is based on the representation and processing of procedural and declarative knowledge, it is imperative to deepen the distinction between these two forms of knowledge.

Declarative knowledge is the material that engineering designers and systems engineers typically handle. It is information that describes

specific system's functions and structures—it represents the "what" of a system. This type of knowledge is typically stored in technical manuals, today in databases or knowledge bases that can be retrieved and used as needed throughout the life cycle of a system. Declarative knowledge representations often include ontologies and semantic networks.

Procedural knowledge encompasses understanding the processes and procedures involved in performing a task and producing an activity. It represents the "how" a human–machine system supports the execution of a task. Furthermore, procedural knowledge processing requires a thorough understanding of the task in multiple contexts, including nominal and off-nominal situations. It is typically represented using free-form scenarios,[49] UML/SysML activity or sequence diagrams,[50–52] business process models and notations (BPMN),[53,54] and procedural knowledge descriptions such as iBlocks.[55]

The HSI-centered combination of procedural and declarative knowledge is crucial in developing an STS—it represents the "why" of a system. However, why a system is designed does not preclude the way human operators could repurpose it.[56] PRODEC proposes a framework to associate these two types of knowledge to support the discovery of STS's emergent properties at work using HITLS and activity analysis.

2.1 | Declarative knowledge modeling and processing: a system as a representation

Declarative knowledge of a system is used during its design phase to define its components and how they are assembled within its environment. As a metaphor, characters and their environment are defined as systemic agents and objects in a novel. This is the declarative knowledge of the novel. Today, digital engineering makes it possible to use such declarative knowledge to virtually define a system as its digital twin. It is, therefore, possible to carry out virtual HCD since digital twins can be used to run HITLS and offer an experimental playground to observe and further analyze the activity of the various human and machine agents involved. Unlike technology-centered design approaches based on task analysis, digital engineering enables human and machine agents' activity observation and analysis during design.[16,44] Indeed, digital twins of the targeted system can be tested and incrementally made tangible toward a targeted concrete system.[26,36] As human-centered system engineers, we understand the importance of designing systems with a strong HSI approach. It requires a human-centered declarative view of the agents involved (i.e., the definition of their roles and tasks), where we understand their goals, functions, and capabilities socio-cognitively, whether humans or machines.

The underlying mechanism and necessary resources for humans to understand and operate the system are also essential. This approach is anchored in socioergonomics,[36] which is based on the TOP model (i.e., concurrent design and development of technology, organizations, and people's competencies).[26] From a multi-agent perspective, PRODEC

[†] Human-centered design started at Stanford University in the late 1950s.[47] It continued to develop in the HSI Community, but we had to wait until recently to see the development of HCD in sociotechnical systems.
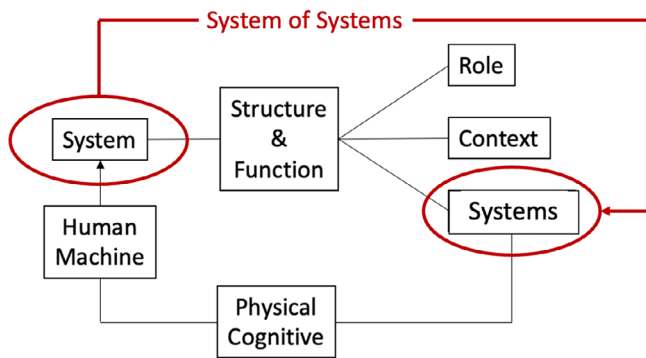
FIGURE 3 Recursive definition of a system.

FIGURE 4 A system of systems in the context-system hyperspace.

supports the definition of human and artificial agents in terms of what they do and their roles, contexts of operations, and available resources.

PRODEC includes cognitive function analysis (CFA)[55]; that allows the elicitation of functions using task and activity analyses concurrently, the decomposition of these functions, and their systematic categorization. CFA is a human-centered declarative modeling method based on operations experience. The grouping and classification of elicited functions allow us to assimilate a smaller number of more generic functions to define their contexts of validity and their necessary resources. CFA can be done directly from interviews with experts and observations of human and machine agents/systems at work or in HITLS. It becomes very complex when there are many agents/systems interacting with each other. This is why PRODEC also includes SBD that supports scenario developments subsequently used in HITLS to observe relevant activities and derive functions from observed activities. This process results in emergent functions associated with various agents/systems (structures) that link prescribed tasks to performed activities. Each elicited cognitive function is declared with a role, a context of validity, and a set of resources that are functions themselves. Once functions have been identified, they are iteratively refined until a satisfactory level of detail has been obtained. Understanding these functions in their use context enables us to redesign systems iteratively and effectively and improve human–system integration.

CFA has been expanded from a systemic perspective, considering systems as representations of both humans and machines.[26,36] Consequently, the HSI approach is recursive (i.e., a system may include humans, and a human may include systems from both structural and functional perspectives). This definition is very important in VHCD, and consequently in digital engineering, because it enables us to model and further implement HITLS to observe activity and further discover emergent behaviors and properties of the system being developed. Figure 3 describes a system as having a structure and a function, each having a role, a context of validity, and a set of resources that are systems themselves. Each system can be physical and cognitive, representing a human or a machine. This constitutes a recursive definition of a system as a system of systems. Therefore, this article expands the cognitive function analysis[55] to physical and cognitive systems analysis.
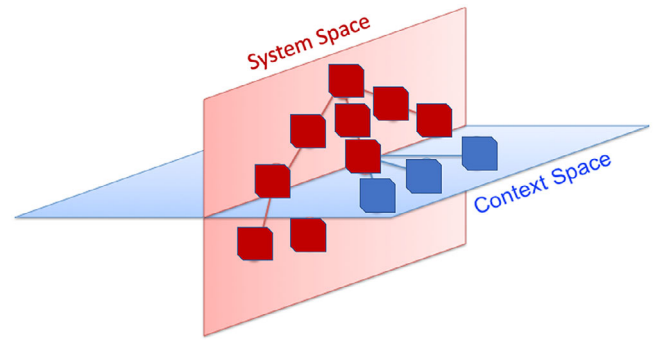
The underlying HSI approach is built around two main dimensions: contexts and systems. Indeed, a system is always defined at the intersection of context and system spaces (Figure 4). A system works within a given context of validity and, at the same time, may be a resource of a bigger system and include other systems. In addition to this teleological representation of a system (i.e., the declarative part), we need to define a system as a logical entity that transforms a task into an activity (i.e., the procedural part). In other words, a task is a system's input, and an activity is a system's output. This is why task and activity analyses are so important. Comparing the two task and activity flows allows us to identify and understand the systems involved and at different levels of granularity. This is precisely the purpose of PRODEC, that is, to develop procedural knowledge in the form of task and activity flows, to compare them, and to determine and refine declarative knowledge in the form of structures and functions as well as systems and contexts. The notion of procedural knowledge is developed in the next section of this article.

2.2 | Procedural knowledge modeling and processing

Considering design as an artist, the system to be designed can be seen as a metaphor for a play, where characters are human and machine agents or systems whose properties and attributes must be defined from experience and creative processes. Let us start with a metaphor to explain how PRODEC could be used. For example, a *producer* wants to create a *modern version* of the *historic play Romeo and Juliet*, where some parts will be replaced by artificial intelligence (AI) driven actors. In our scenario, all copies of the play have unfortunately disappeared. Some pamphlets presenting the play remain, some *parts*, and some *former actors* have played these roles. How can the *producer* go about creating a new version of this play? First, the producer asks former actors about the play and, based on their stories, begins to recreate the script (i.e., the *scenario*), discovering new *emergent parts*. From the reconstructed script and the testimonies of the former actors, he can begin to get an idea of the setting as it was. The producer asks *former actors* to review the reconstructed script to fine-tune his vision. Finally, after memorizing most of the script and setting, he can begin to understand the underlying motivations of the actors behind

their actions. The producer then discusses with a foreign company performing Romeo and Juliet. Interpretation is key, and a previous producer may have chosen a specific interpretation for his version of the play. But this is not the only interpretation or the best one. Having a good vision of Romeo and Juliet, the producer decides to modernize it, to adapt it to the current context, where society and technology have evolved. It would not have been possible to change one part of the scenario without understanding the motivation of each part of the context. Romeo will now be a state-of-the-art AI actor—the producer is not creating fiction—and Juliet to a more modern depiction of a woman. Some of Romeo's historical features can easily be carried over into the new version. Others are impossible and must be reassigned to other parts for the story to hold together. The *producer* proposes new scripts and organizes "rehearsals" with *new actors*. He observes the *activity* of the human and machine characters, the flow of the play, and what *emerges* from the interaction of the parts. Based on practical experience with the play, the producer iteratively modifies the scripts, roles, and setups to improve the production.

From this theatrical metaphor, let us provide a more generic framework and process. We start with eliciting the expert's experience, that is, procedural knowledge (e.g., how actors played in the above metaphoric scenario). Knowledge elicitation has been studied and practiced for a long time.[57–59] Experience comes primarily from operations people who typically tell stories about what they have experienced through various interview techniques or, if recordings are available, self-confrontation interviews.[60,61] Indeed, even if this knowledge elicitation approach provides a solid framework, it does not capture tacit non-verbal dimensions people are involved in. Only observations of operations, either recorded or directly in the field, enable access to these dimensions. For example, in aeronautics, flight tests support such observations. This experience-based approach to engineering design and SE must be cross-fed by participatory design[62] and creativity.[63] This is typically done through vivid sketches, storyboards, and stories-based cartoons, provided by subject matter experts (SMEs), which lead to animations, simulations, and so on. Art designers are critical partners who create media that support collaborative work (i.e., mediating representations shared by the entire design team, such as digital twins‡).

According to Georgeff,[64] most of the world's knowledge comes from procedural knowledge. Procedural modeling plays a crucial role in the design and development of complex STSs. To effectively model these complex systems, we need procedural modeling that supports multi-levels of details in different dimensions, such as space, time, and other independent dimensions. These dimensions can vary greatly along the design process and need to be accommodated for a successful system design.[6] One critical requirement for procedural modeling is the support of natural and artificial agents. The system needs to consider the tasks and functions performed by each agent and the material and cognitive resources required to perform those tasks effectively. Additionally, modeling must support multiple contexts, including AS-IS scenarios considering what already exists, TO-BE scenarios considering what is anticipated in the future, and various phases of system operation and maintenance, as well as normal, abnormal, and emergency situations.

Several systems engineering models have been developed to describe procedural knowledge, such as SysML/UML activity diagrams or sequence diagrams. However, we have tried to use these models and found that they offer only limited support for context. BPMN diagrams, a standard for business process modeling, provide a more comprehensive solution by supporting procedural knowledge elicitation and graphical formalization. With BPMN diagrams, each agent has its lane, where tasks and their relations with other agents are described. BPMN diagrams enable the description of procedural information with different graphical elements in the form of scripts, episodes, sequences, and so on, which mixes how agents interact with each other—it is a program or a routine in the computer science sense, as SysML, for example, is better suited for the description of declarative information.

2.3 | The procedural-declarative integration loop

Unlike conventional engineering approaches that start with declarative constructions of a system and then procedural tests, PRODEC method starts with procedural scenarios development based on existing experience in the field at stake, provided by subject matter experts (SMEs), and develops declarative configurations of the system by incrementally integrating emerging properties discovered in human-in-the-loop simulations.

More formally, the PRODEC process starts with the elicitation of procedural knowledge in the form of an iBlock-based AS-IS analysis based on operations-dedicated SMEs (O-SMEs) experience, followed by the elicitation of underlying declarative knowledge in the form of an AS-IS analysis of the system leading to the creation and development of a TO-BE virtual prototype,§ based on design-dedicated SMEs (D-SMEs) experience, and subsequent HITLS. This is how an activity analysis can be performed in PRODEC using the iBlock formalism (Figure 5). Consequently, task and activity can be compared using both iBlock diagrams.

PRODEC is based on the requirement that the different agents/systems developed, in terms of functions and structures, human and/or machine, must be understood and therefore declared in terms of activity produced and not only of tasks to be performed. Operations context (e.g., context of use for a tool) is crucial and requires testing many scenarios to evaluate the safety, efficiency, and comfort of the solution to be chosen. For this reason, the loop shown in Figure 5 requires enough iterations to obtain satisfactory declarative knowledge (e.g., systems require enough tests and operations time to be certified). In other words, task-activity workflow expressed in terms

---

‡ Our vision of digital twins will be provided further down in this article.

§ An AS-IS process is defined (or modeled) by its "sub-processes, workflows and activities, as it happens currently in the organization." A TO-BE process is modeled by its "targeted performance and requirements of the organization – related to the system being designed – thereby also addressing the issues faced with the current process."[65]
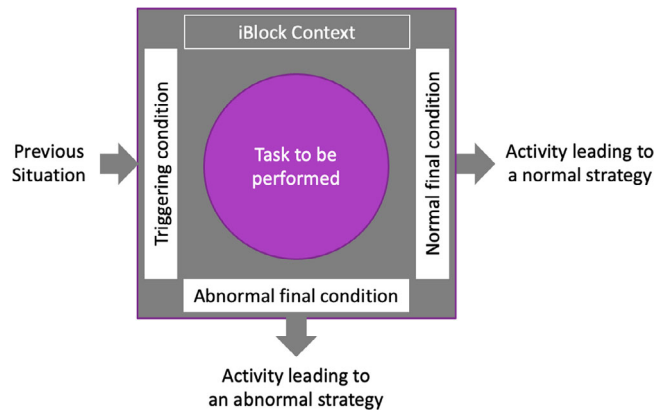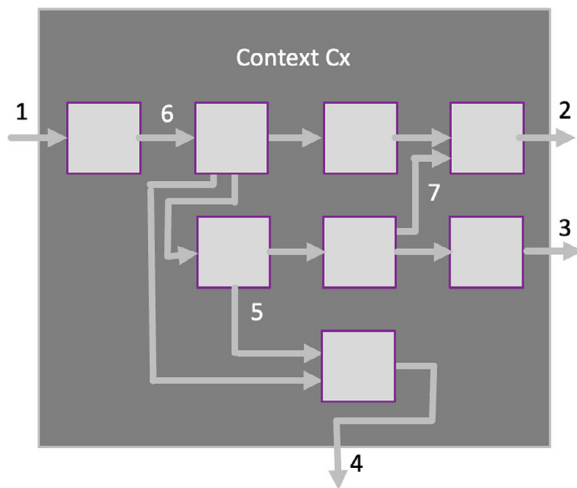
FIGURE 5 iBlock representation.

FIGURE 6 Recursive refinements of an iBlock.

of iBlocks¶ and, more specifically, contexts of iBlocks (i.e., procedural knowledge as formalized in Figures 6 and 7) enables us to identify already existing or emergent systems (i.e., declarative knowledge as formalized in Figure 3) producing these contexts of iBlocks.

Figure 8 shows the human-centered digital engineering involving the iterative tangibilization of the VHCD process, in which we start by gaining experience from SMEs. This approach is based on the development, use and refinement of virtual prototypes, called digital twins of the system being developed. It requires incremental development of virtual human and machine agents, as well as a "Control and Management Space" (i.e., a kind of user interface, or a control and management room). After a few iterations, the design team, where all possible SMEs are included (i.e., involved in a kind of participatory design), provides novel appropriate design solutions that satisfy the requirements constructed by D-SMEs from experience feedback of O-SMEs. This iterative (agile) process continues until a satisfactory result is found. After the second iteration, some parts of the designed solu-

tion can be redesigned and incrementally made tangible. The last step is when all system components are both physically and figuratively (i.e., cognitively) tangible[36]; it is called "Tangible Human-Centered Engineering."

Human operators can provide experience on what they know and need, but they do not know how a new system could be developed unless they are also engineering designers. Indeed, they know about operations using current technologies in current organizations but do not know about something they never experienced. Operations people usually describe the way they do things by telling stories. In other words, they describe their experience procedurally (e.g., resulting descriptions of their know-how can be done in timelines and graphs of events). Storytelling has been extensively investigated in human–computer interaction design.[66]

Summarizing, PRODEC method is implemented using the process presented on Figure 9. There are two main sources: (1) elicitation from operations experts; (2) direct observation. Knowledge elicitation (e.g., through interviews**) from experts leads to task analysis, during the early stages of engineering design. Sometimes, verbalizing is unnatural, so interviews help elicit knowledge. It allows a design team to explore the system's context of use, user's needs, how a system can be evaluated, information considered by a human operator during the execution of a task, or mental representations of the person being interviewed, for example. Direct observation, followed by activity analysis, is good if you can access existing systems easily. However, for large complex systems (e.g., space systems), operations can be difficult to monitor if they occur on multiple sites; timewise, operations can be long or short; organizationally, several agents are interacting; other sources of complexity should be investigated (e.g., operations can be diffuse, rare, or in the background). Observation is a field method that involves collecting observable data from human operators. The data collected are behaviors, verbalizations, and interactions between colleagues and with manual and technological tools. Observation enables us to understand the operators' working environment without interfering. It is the best way to get away from individual interpretations and access non-verbal behaviors that people are not always aware of. Direct observation is typically performed in the human-in-the-loop simulation. The activity must then be confronted, condensed, compiled, and consolidated to redesign tasks from which we deduce the function.

Operators in simulations can either rely on existing procedures or generate new procedures through learning by doing. Those who follow established protocols benefit from efficiency, consistency, reliability, and adaptability. On the other hand, operators who generate new procedures can foster innovation and explore alternative solutions. However, this approach can also lead to increased cognitive load and the need for more standardization.

¶ The iBlocks and BPMN diagrams presented in this document are barely legible but are provided to show the qualitative complexity of real-world examples of task and activity analysis support.
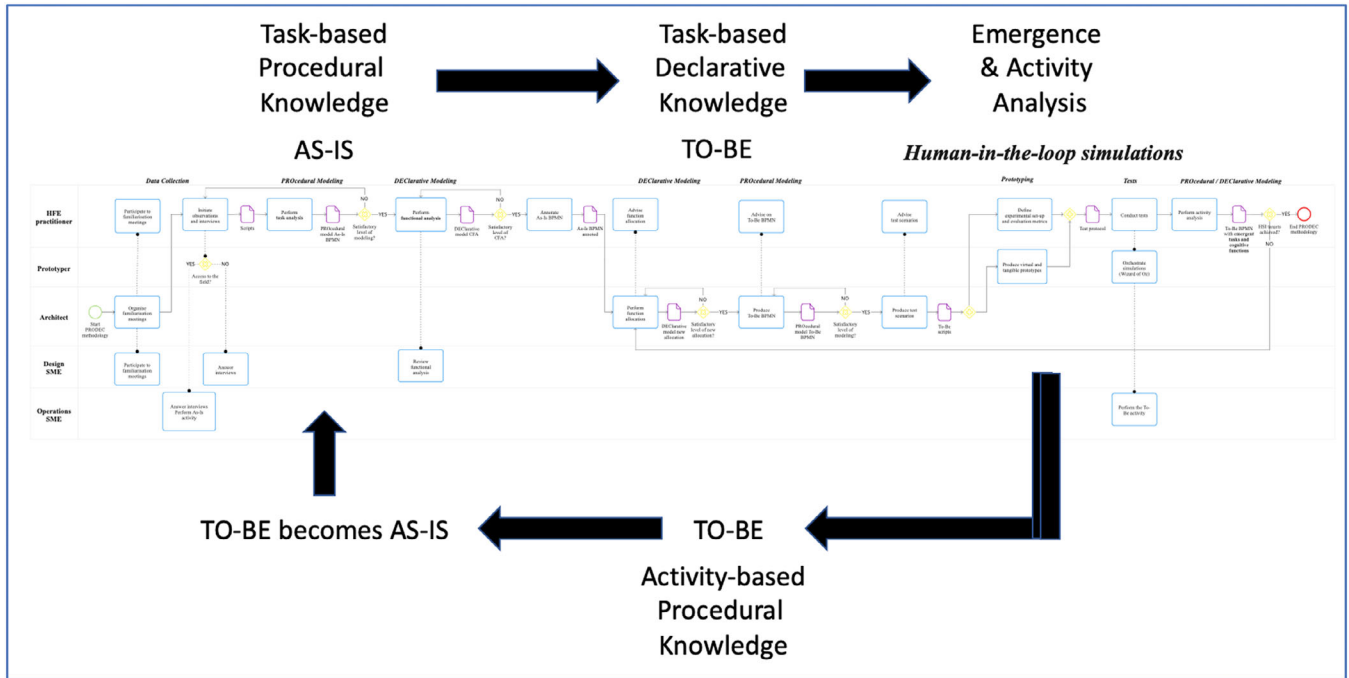
** It is recommended to conduct semi-structured interviews (i.e., the interviewer conducts the interview following an interview guide). Human factors specialists must do some preliminary research beforehand, to get familiar with the activity concerned (e.g., terminology, tools, techniques) to build the body of the interview guide. Interviews should also be done after the observation phase and will allow human factors specialists and human operators involved to comment on observations. Recordings could be used to carry out self-confrontation. Observations and interviews enable to obtain detailed descriptions of tasks performed, actors involved, and tools used, as well as time and geographical aspects, as well as possible disruptions.

FIGURE 7 Overview of the PRODEC knowledge acquisition framework on an iBlock diagram.

2.4 | PRODEC human factors metrics

At this point, it is important to provide a set of appropriate metrics that enable a design team to help discover emergent properties discovered during HITLS. From a human factors point of view, three main dimensions have been considered in PRODEC: cognition; maturity; and life-cycled system evolution. These evaluation dimensions will likely evolve as PRODEC is used and developed.

From a cognition point of view, three main metrics are typically used in PRODEC: situation awareness (SA); decision-making (DM); and action-taking (AT). The underlying model is called SADMAT. SA assessment is based on Endsley's work and evaluation techniques.[67] DM assessment is based on Wickens's work and evaluation techniques.[68] AT is considered as risk taking, which involves preparation, attention, concentration, accuracy, stability, and other factors.[69] Two other global metrics are also considered, workload, and operational performance.[68,70] Finally, PRODEC includes two socio-cognitive metrics, trust and collaboration,[17] that deal with the multi-agent nature of PRODEC.

Regarding maturity, considering the TOP model,[26] the PRODEC approach considers three types of scales: technology readiness levels (TRLs)[††]; human readiness levels (HRLs)[71]; and organizational readiness levels (ORLs).[36] Indeed, the articulation of these three types of maturity metrics is crucial for enhanced HSI, as already discussed.

There are three metrics that allow to better understand the evolution of a sociotechnical system during its entire life cycle: resource commitments (RC), design flexibility (DF), and system knowl-

edge (SK).[26] In the technology-centered SE approach, RC drastically increases from the very beginning of the life cycle of a system to saturate very quickly, which does not leave many DF opportunities for engineering teams to correct possible design flaws. They incrementally learn about the system by testing it, with a substantial growth when the system is fully developed and holistically tested. Conversely, in human-centered SE design approach, such as PRODEC, SK can grow earlier and faster using HITLS that allows discovery of emergent properties of the system being developed in virtual environments. At the same time, DF remains high during the digital engineering phase (i.e., using virtual prototypes), which leaves many opportunities for engineering teams to correct possible design flaws, and commit as late as possible on hard resources (i.e., resources that cannot be easily modified or changed).

3 | A REAL-WORLD CASE STUDY

3.1 | Problem statement

Consider designing a new remote maintenance system for an oil and gas platform where the human operators currently performing routine maintenance tasks are to be replaced by mobile robots for emergency operations. The goal is to create safer, simpler, lighter, and more cost-effective facilities with increased reliability and reduced maintenance. The project involves rethinking the whole architecture (Figure 10) and organizing the platform around robots, which will be remotely supervised by a robot panel operator in the operation room. This room would be next to or replace the existing control room that supports the supervision of the oil process, which would also become remote. These

---

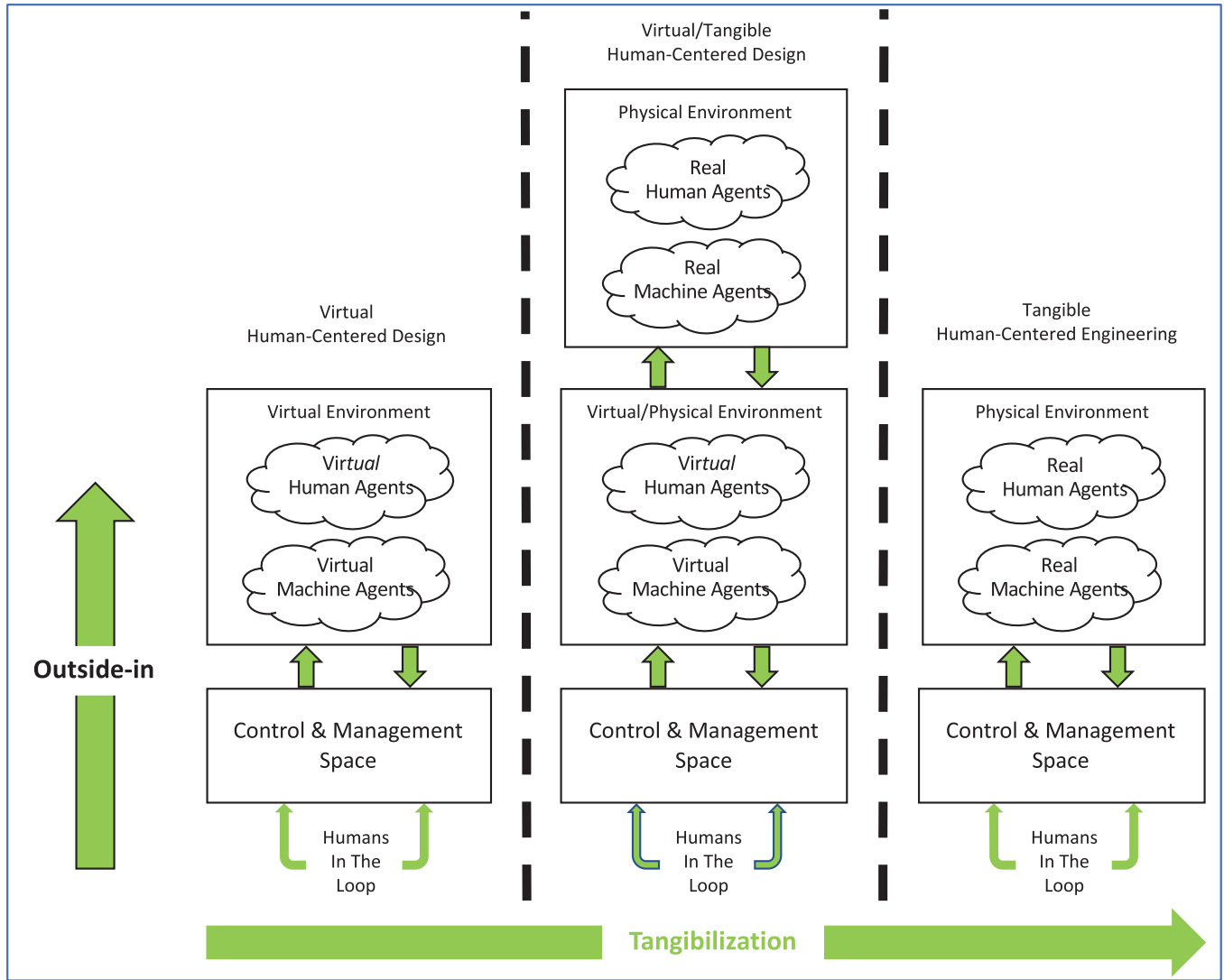†† https://www.nasa.gov/directorates/heo/scan/engineering/technology/technology_readiness_level.

FIGURE 8   Human-centered digital engineering based on tangibilization of VHCD process.
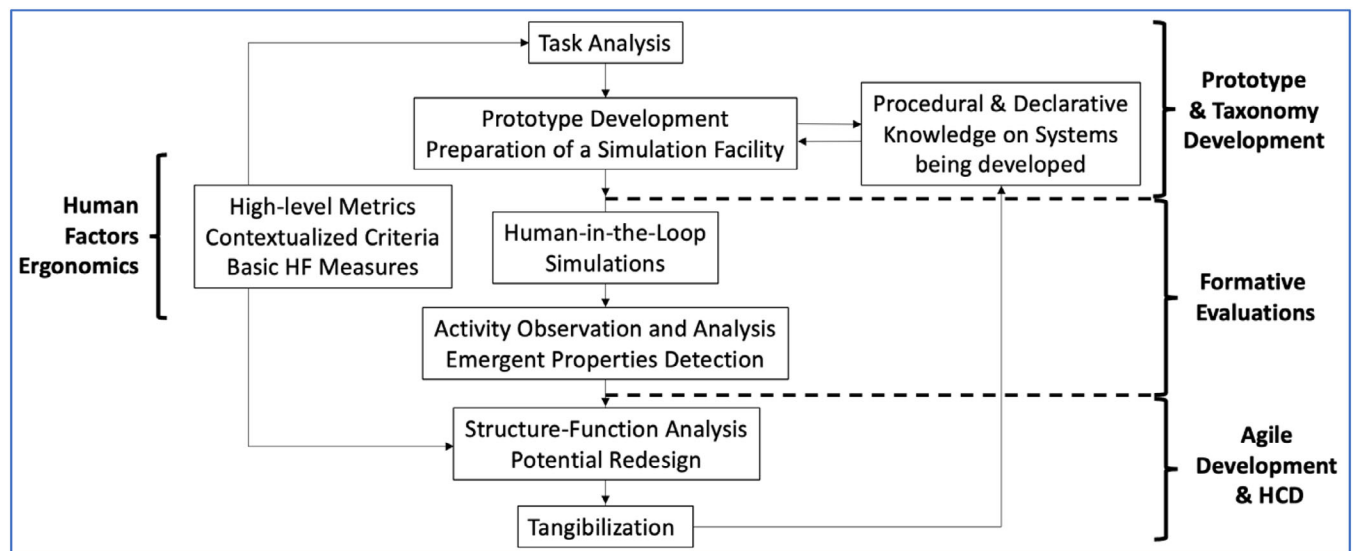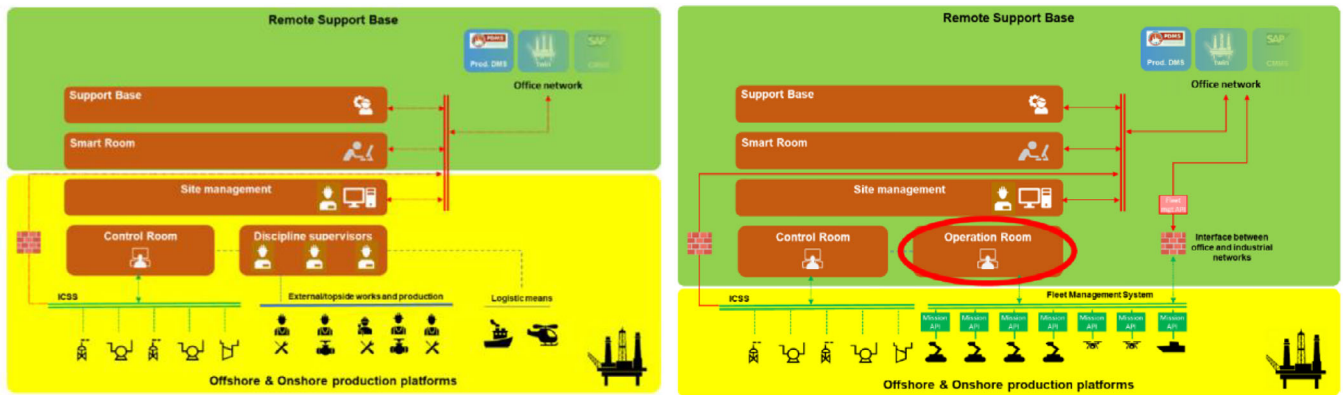

FIGURE 9   PRODEC process.

FIGURE 10 Current (left) and future (right) organizations of platforms.[72]

new platform facilities would no longer require a continuous human presence. Human intervention would be limited to short maintenance campaigns. The project aimed to study the impact of the introduction of robotics in the management of operations, which will be carried out from the platform's remote control room and operation room. The objective was to specify the organization of the new operation room entity and to identify the skills that the operators of this room should have.

We developed a virtual (digital) simulation of the system (i.e., the oil-and-gas platform and the fleet of robots) from a preliminary task analysis with D-SMEs. A HITLS facility was then developed and used to play various kinds of scenarios. Playing these scenarios involving O-SMEs, generated activity was observed and further analyzed involving D-SMEs. Note that this activity analysis typically leads to discovering emergent properties, typically in the form of functions and structures, that can be physical and/or cognitive.[26] Emergent properties were then considered for the re-design of the system in subsequent more tangible simulations of the system. The process of tangibilization in VHCD involves transforming virtual human or machine agents into physical ones. This can involve either physical development of the system (replacing a virtual robot with a physical one) or modification of functional parts for improved cognitive processes (such as situation awareness, decision making, and action taking). Tangibilization is thus both physical (replacing a virtual agent with a physical one) and cognitive (enhancing cognitive processes).

3.2 | From procedural modeling and data collection to AS-IS scenarios

In our experiments, our pool of scenarios regarding the control room was selected to apply the methodology: from routine activity, with some manipulation and coordination for gas detector calibration, to more complex activity involving many manipulations and coordination between agents for pig launching, the process of sending a scraping element through the pipe system. Gas detector calibration requires coordination between several entities in a typical production organization and involves the following steps:

- From the computerized maintenance management system, a tool dedicated to preparing and scheduling maintenance activities that are ordered in a buffer of tasks in 5-week look ahead schedule.
- The production site management team conducted a job risk analysis to determine the date of work and the mitigation measures to be implemented for execution. This analysis was performed as a team effort involving the production, maintenance, and safety teams.
- The intervention was then conducted after a series of final site checks, a workplace conversation between those involved, and the implementation of mitigation measures, including safety waivers identified by a control systems specialist. All these tasks were coordinated between the site and the control room in real time.

In practice, pig launching requires a permanent communication link for process synchronization between field operators acting on the pig launcher and the control room operating the plant. In addition to this coordination, the field operator had to evaluate and assess the site conditions to launch the pig safely. A typical sequence of operations (i.e., function executions) for existing installations was:

- Pig launching is initiated by the control room, informing field operators.
- A first site inspection for coactivity risks and site integrity is performed on site, before the pig launcher is purged and flushed to be ready for opening. This requires synchronization between field operators and panel operators for safe operations.
- After pig insertion, local valve operation is conducted simultaneously with process change controlled by the control room to launch the pig in the pipeline.
- After pig is launched the pig trap is again put in condition for opening and checked with a close synchronization between the field operators and the control room.

HSI practitioners followed a familiarization phase conducted by a system architect to understand the general organization of operations on oil and gas platforms. This preliminary research effort allows the preparation of interviews. Interviews were conducted with D-SMEs (former operating authority, production supervisor, production panel
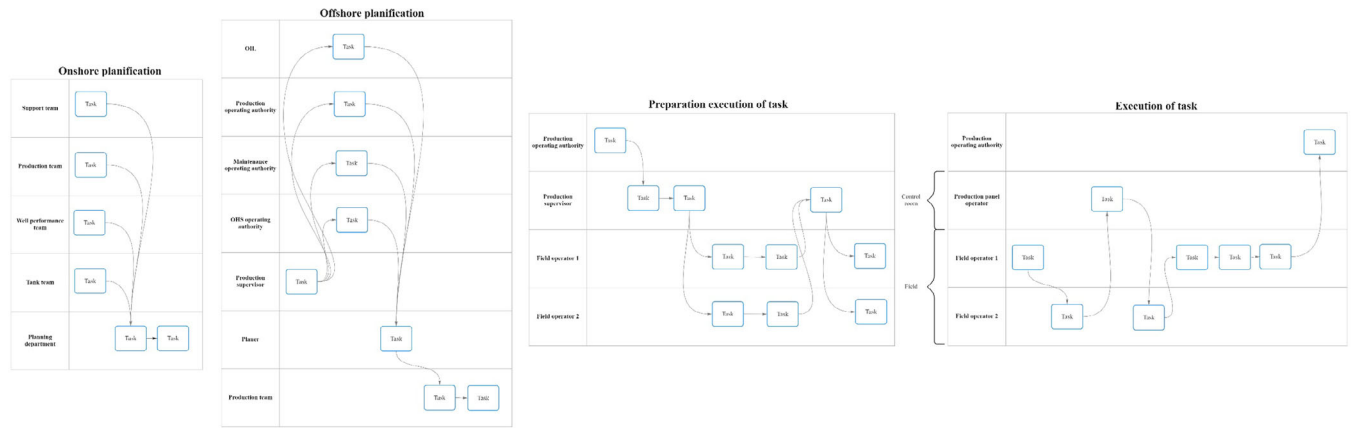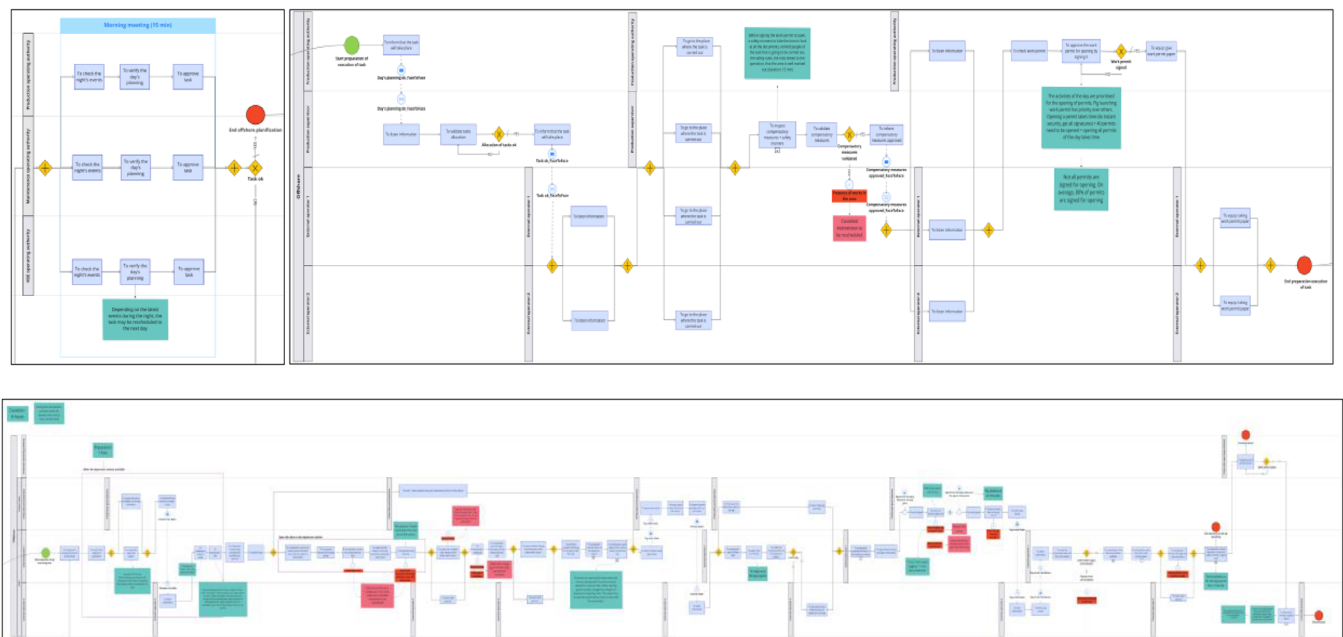
FIGURE 11 Macro iBlock of pig launching.

FIGURE 12 AS-IS BPMN of pig launching.

operator). Interviews were transcribed *verbatim* (word for word), and an initial modeling of the intervention in iBlock/BPMN diagrams was carried out. Following this, three iterations of interviews took place using the BPMN as a support. The aim was to review the BPMN diagram with the D-SMEs to complete it (errors, missing tasks, lack of detail) and explain understood elements. Following the different interview phases, a BPMN diagram describing how a pig launching is currently performed was formalized. The Procedural Model had multi-levels of detail. From a macro viewpoint, it was divided into four main parts that define four contexts of iBlocks (Figure 11): Onshore planning; Offshore planning; Task preparation; Task execution. In each phase, tasks performed by the different agents were detailed (multi-agent approach).

Several situations were represented simultaneously on the BPMN diagram (Figure 12): the default path (→) represents the process in normal situations. Alternative paths represent abnormal situations. A color distinction has been made between S (black), R (blue), and K (red) level tasks (Figure 13), referring to Rasmussen's model.[73] Red boxes indicate examples of disturbances that may occur during the intervention. The tasks resulting from these disturbances are written on the pink post-its. Explanations and details of the intervention are given on the blue post-it notes. This example shows the complex interactions between humans who are currently required. Considering that in future facilities, nobody will be on-site except robots and that robots will execute tasks currently executed by humans, it becomes clear that a review of the different roles and responsibilities is required.

3.3 | Declarative modeling

Performing a functional analysis, each task in the AS-IS BPMN model was first listed in a spreadsheet, along with the function and actor.
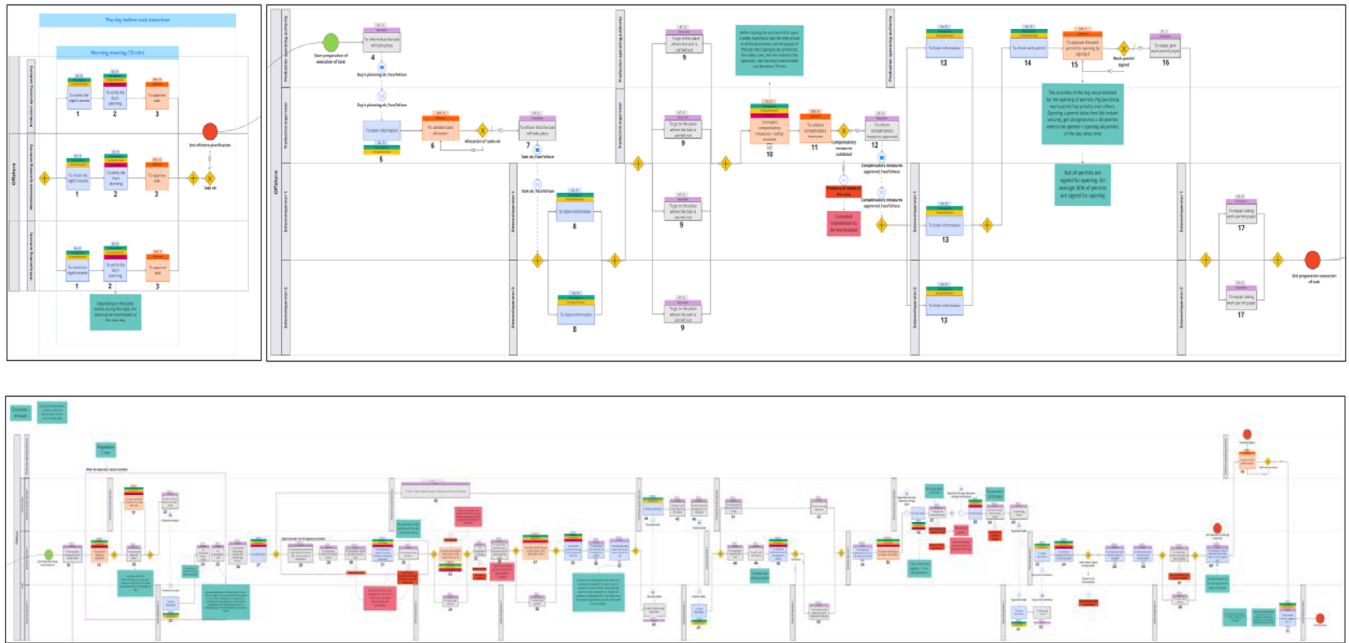
FIGURE 13 Annotated AS-IS BPMN of pig launchin.

However, before starting to describe the tasks, it has been necessary to carry out a preliminary work of definition of the various functions. This made associating each task with the right function possible. Thirteen functions were identified and categorized according to the Endsley and Rasmussen models.[67,73] At the same time, the cognitive functions required to perform the task were determined (Table 1).

A functional analysis was conducted based on day-to-day activities, from the morning meeting to closing the permit and writing the report (specifically, onshore and offshore planning were not considered). Seven agents were involved: external operator 1, external operator 2, production control panel operator, production operating authority, maintenance operating authority, organizational and human system (OHS) operating authority, and production supervisor. The functional analysis was validated with D-SMEs, and the AS-IS BPMN was annotated: for each task, the type of function and the cognitive resources were added (Figure 13).

3.4 | Redesign

We focused on day-to-day activities for the TO-BE modeling because replacing external operators by robots will have the most impact on this part of the process.

3.4.1 | Function allocation

Function allocation between humans and robots was carried out for the tasks performed by the field operator (Table 2). It is based on the type of function and the cognitive resources that the function requires. In addition, certain architects' specifications were considered, such as the

desire not to install additional electronic elements on the equipment, and to modify the current processes as little as possible. For example, the new procedure proposes that most safety-related functions be performed by humans. The tasks performed in the control room are unchanged, and the planning and work permit have not been changed, although they could have been. The proposal is a mixture of what is technically feasible and what can be accepted.

Action functions ("Go," "Apply," "Equip," "Inform," "Insert," "Manipulate") performed by the field operator have been allocated to the robot. In the case of a robot, the function "Inform" becomes "Send." "Approve," "Check," "Document," and "Validate" functions remain performed by control room operators. For SA functions, a robot can only "Receive" ("Listen") information. This function requires perception, comprehension, and projection of information. The functions "Inspect" and "Verify," performed by field operators. It was, therefore, chosen not to assign these tasks to the robot but to create a new human agent. The robot panel operator.

The robot is an instrument/relay between the field and the remote operators. For example, for the pig launching, when the door of the launching station is opened, the robot does not inspect the station's interior, as the operators currently do. The robot takes pictures of the inside of the station and sends these pictures to the robot panel operator in the operation room. The robot panel operator verifies the interior of the station with the photos sent by the robot and validates, or not, whether the intervention can continue. Numerous checkpoints take place throughout the intervention, during which the robot panel operator must "Verify," "Inspect," and "Validate" the information (photos, data) sent by the robot. The information between the robot and the operation room is transmitted via the fleet management system (FMS). Figure 14 summarizes the allocation of functions between the different agents.

TABLE 1 Useful functions to perform a pigging operation on an oil & gas platform.

Function	Definition	Type	Cognitive resources
Go	Move around	AT; S	Execution
Apply	Follow procedures or rules	AT; S	Execution
Approve	Give approval because one has the competence and authority to do so	DM; K	Selection
Check	Examine, to look for information	SA; R	Perception Comprehension
Document	Carry a certain amount of information on a support	AT; S	Execution
Listen	Pay attention to what someone is saying to hear and understand it	SA; R	Perception Comprehension
Equip	Provide themselves with the necessary equipment for a given activity	AT; S	Execution
Inform	Transmit, communicate information	AT; S	Execution
Insert	Enter information into a system	AT; S	Execution
Inspect	Observe carefully and thoroughly good functioning condition, check and verify	SA; K	Perception Comprehension Projection
Manipulate	Handle in view of an operation	AT; S/R	Execution
Validate	Confirm, make or declare valid	DM; K	Selection
Verify	Ensure compliance of parameters	SA; R	Perception Comprehension Projection

Abbreviations: AT, action taking; DM, decision-making; K, knowledge; R, rules; S, skills; SA, situation awareness.

TABLE 2 Allocation of functions between human and robot.

External operator functions	Allocation
Go	Robot
Apply	Robot
Listen	Robot → **Receive**
Equip	Robot
Inform	Robot → **Send**
Insert	Robot
Inspect	Human
Manipulate	Robot
Verify	Human

3.4.2 | TO-BE scenario

With the declaration and allocation of available functions, multiple robotic scenarios were designed with different levels of automation. Several iterations were performed to arrive at the BPMN diagram of Figure 15 validated by the architect.

Establishing this robotic scenario made it possible to determine which type of robot (e.g., inspection, simple manipulation, complex manipulation) should be used to operate. A simple handling robot can do the job since the pigging operation essentially consists of closing and opening valves.

This scenario allowed us to determine some purposeful specifications for the robots, such as each robot must be equipped with a camera to take pictures, film, etc., a light, and a gas sensor (equivalent to a portable gas sensor). Each robot must be able to identify valves, open and close valves, and know the state (open/close) of the valves. As iterations progress, these specifications will be completed and refined.

3.5 | Human-in-the-loop simulation

The proposed new process must be tested in a simulation as accurately as possible to assess its practical feasibility. The simulation was conducted on a robotics development platform where a simulated plant was created with the capability to test robots and conduct gas leak detections. As there were no mobile robots capable of physically interacting with the installations, the simulations were carried out with an older generation of remotely operated robots. The plant was also a model of a process plant free of hydrocarbon where pig launching can be simulated without any risks. A remote operation room was set up. It is fitted with a remote-control system for operating the robot and a simulator for the process control system of a real plant. An observation team was present to analyze the simulations and make conclusions. Experienced field personnel were mobilized to perform the simulation in the most realistic way possible.

3.5.1 | Prototyping and Wizard-of-Oz

In HSI, the Wizard of Oz method enables a human operator to interact with a machine without knowing that the responses are generated by a human rather than a machine by having someone behind the scenes producing the machine's activity. The experimental setup for the test is described below and shown in Figure 16. It uses real elements to define the equipment, control screens, processes, etc.

The following set up was developed and used:

- In the field, one installation served as the departure and arrival station of the pig with associated equipment (e.g., closed drain line, valve, etc.). This choice of installation and equipment matching was made with D-SMEs' help. The Tracker robot was remotely operated
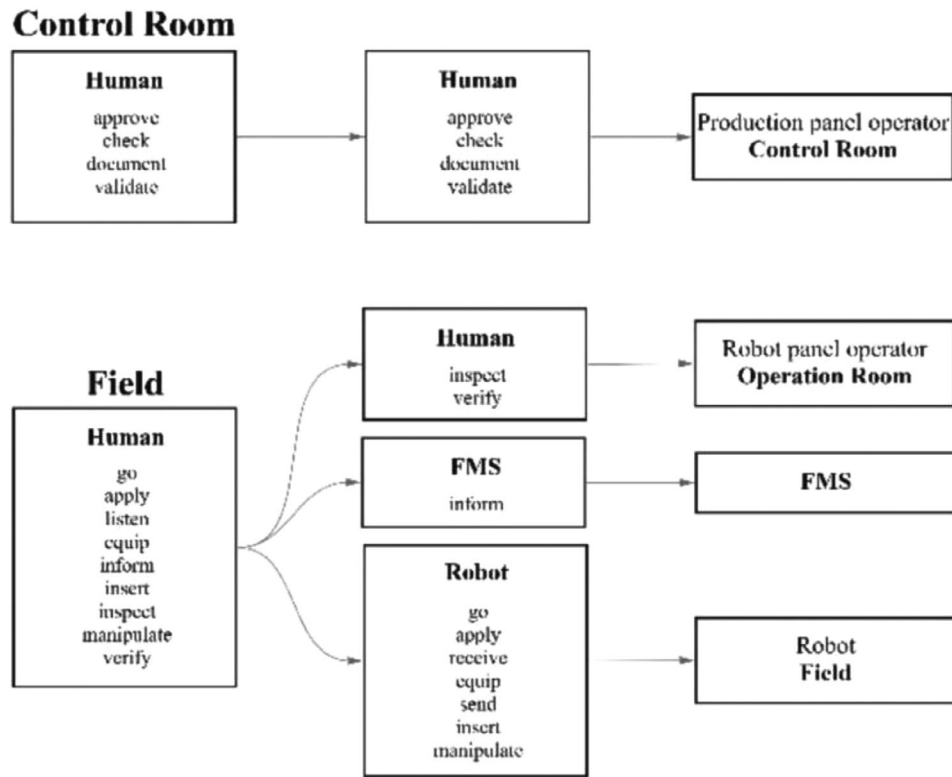
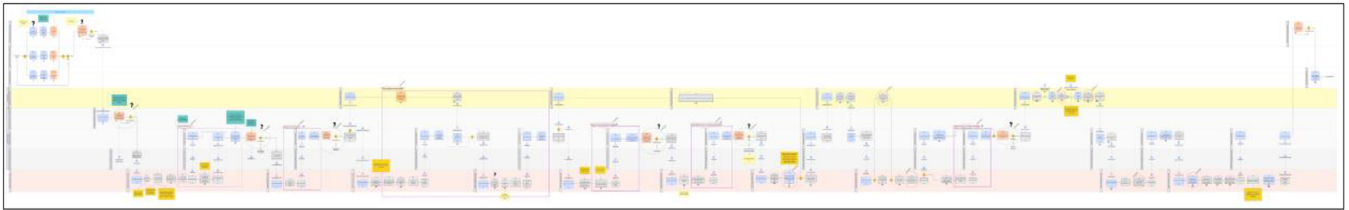**FIGURE 14** Allocation of functions between the different agents.

FIGURE 15 TO-BE BPMN of pig launching.

from the operations room by the robot panel operator. The robot was used for image-taking tasks (i.e., information taking) and manipulation tasks (e.g., valve opening, door opening). Simply put, a person in the field can perform the handling tasks instead of the robot. This person also performs additional tasks to ensure consistency of operations, such as removing the pig from the departure station when it has left using the Wizard of Oz principle.[74] This person was in contact with the operations room by radio to follow the progress of the operations.

- In the control room/operations room, one person played the role of the robot control panel operator, and one person played the role of the production control panel operator. A GoPro camera filmed the behavior of each person. The operator of the robot control panel had two screens in front of him: one screen with the tracker robot control software for remote control and a screen providing a 3D

Unity-based model of the experimental setup (digital twin). This tool was used to help the robot control panel operator when he controlled the robot, to know where it was located at the equipment level.

- The production control panel operator faced two screens. The primary screen presented an interactive model of the process control. The secondary screen displayed a PowerPoint document. Before operations started, a morning meeting was held between the robot panel operator, the production panel operator, and an operating authority. Each day, the purpose of this meeting was to verify the schedule and review the list of work permits. The objective was to manage potential problems of coactivity. If no problems were detected during this meeting, pig launching could begin. The work permit preparation phase for the pig launch was completed before the morning meeting began.
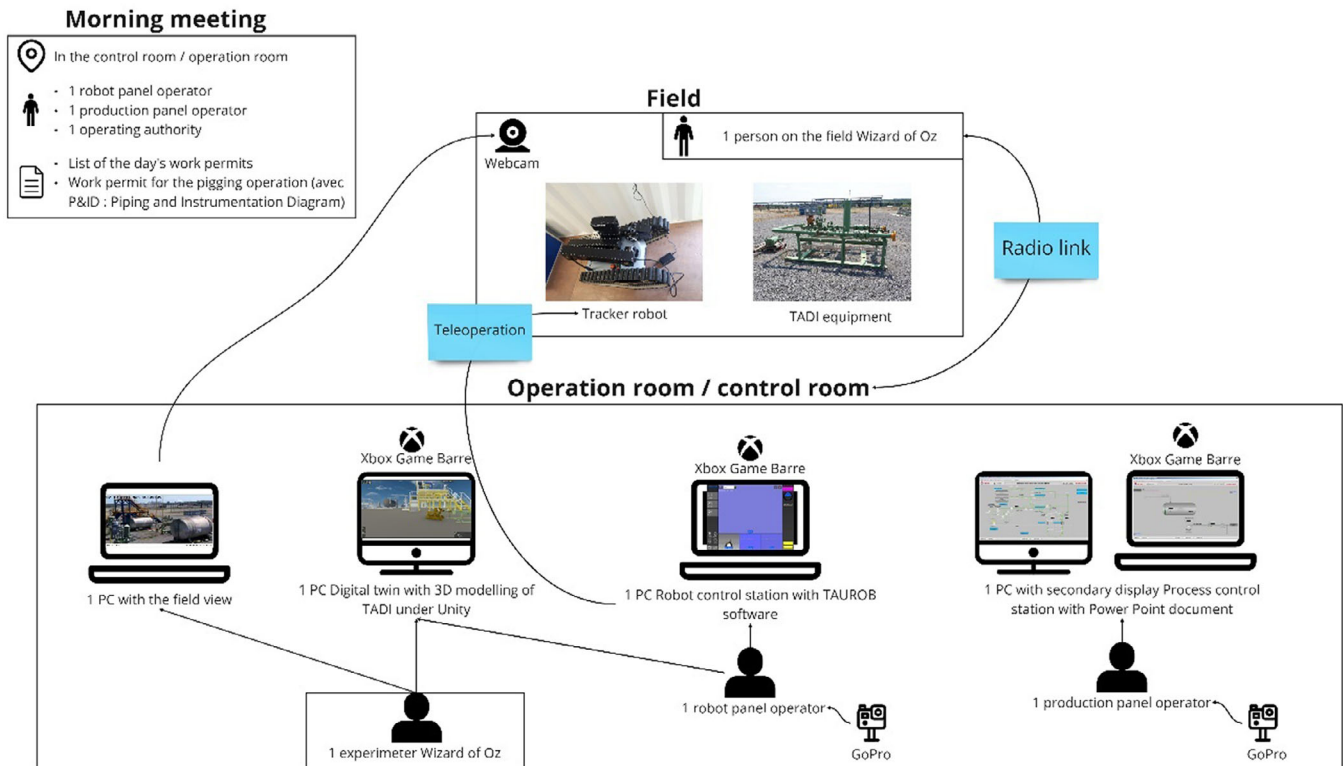
FIGURE 16 Diagram of the experimental set-up for the test.

3.6 | Activity analysis and evaluation

3.6.1 | Activity analysis

The test carried out was the subject of an activity analysis. The objective was to model the activities carried out by the robot panel operator and the production panel operator in BPMN diagrams to understand their various interactions. The BPMN diagram made it possible to count the number of tasks and activities carried out by each participant and the number of interactions they had. An analysis grid was designed to complement BPMN diagrams to study all the interactions during the test. This grid supported the acquisition of subjective and objective measures related to different collaboration criteria for each observed interaction. The criteria were effectiveness, efficiency, and feedback quality. The objective was to be able to qualify the collaboration for each interaction: good collaboration (green), collaboration with defects (orange), and no collaboration (red). This grid was improved with each iteration and was broken down into three parts:

1. The context (video editing time indicator; type of interaction: interaction from the robot panel operator to the production panel operator, interaction from the production panel operator to the robot panel operator; nature of information: transmission of information, request for information, response to a request; BPMN task to which the interaction relates).
2. Collaboration criteria (efficiency estimated from experimental analyses of the total time of the interaction between the two participants, the time a person starts to give information to another, the time the other person has finished giving his answer; effectiveness [quality of information] estimated from reasons and number of repetitions, rephrasing, tone, and so on; feedback from an interlocutor).
3. The experimenter's remarks.

3.6.2 | Scales and questionnaires

In addition to the analysis grid, other collaboration criteria were studied via standardized scales, questionnaires, and semi-structured interviews. The criteria selected were awareness (situation awareness), effort (mental load), ease of use (user evaluation of the different components of the system), and engagement (user involvement).

Situation awareness was assessed using the standardized Situation Awareness Rating Technique (SART).[75] This scale assessed nine dimensions of situation awareness: situational instability, situational complexity, situational variability, level of arousal, concentration of attention, division of attention, level of cognitive ability, amount of information, and situational familiarity. The nine dimensions will be assessed on a scale of 1 (low) to 7 (high).

The individual mental load was assessed via the standardized NASA-TLX (NASA-Task Load IndeX) scale.[76] This scale was based on the evaluation of six dimensions related to mental load: mental demand, physical demand, time demand, performance, effort, and frustration. The six dimensions were evaluated using a scale of 1–20.

Task and team mental load were assessed via the Team Work Load Questionnaire (TWLQ).[77] This questionnaire was divided into questions 1–6 on assessing mental workload about the task and questions 7–12 on assessing team mental workload. The questions were rated on a scale of 0 (very low) to 10 (very high).

The usability of the robot control interface was evaluated via the standardized System Usability Scale (SUS).[78] Usability is the degree to which identified users can use a system to achieve defined goals with effectiveness, efficiency, and satisfaction within a specified use context. This questionnaire included 10 questions on a scale of 1 (strongly disagree) to 5 (strongly agree), alternating between positive and negative questions. The involvement of the human operators was assessed through a post-test interview.

3.7 | Results

Initial results from the simulation highlight the following aspects. Modifications were limited to delegating physical actions to the robot but keeping all the decision-making to the operators in the control room. Under these conditions, communication within the team tended to remain unchanged from the traditional operating philosophy. In addition, efficiency tended to increase because the limitation given by portable radios was removed when everyone is in the same room. It also seemed to retain the same frustrations as current human operators.

An example can be given by the time to wait while the robot moves from one place to another, comparable to what happens when a field operator must move. However, limiting decision-making to humans was not optimal. Today, digital tools have made it possible to delegate more activities and decision-making processes to the machine, such as:

- Task scheduling considering coactivity surveillance in an unmanned context.
- Identification of leaks and loss of containment by robot sensors.
- Sequencing of robot activities with adjustment of safety sensor overrides.

But then, trust in the system and, thus, users' acceptability were compromised. One of the objectives of these simulations was to identify the right limit in delegating functions to the robotic system.

Conducting the study and simulations before developing the first robot prototype and using the robotics test center led to too many assumptions. During the simulations, the actors playing the role of operators had to deal with too many deviations between what was played and what would be the real case. Consequently, the next development for applying this methodology will include building more accurate robot simulators. In the future, we plan on playing the HTILS with a 3D digital twin that integrates the model of existing plants and advanced design for robots as early as its engineering is mature enough but well before the prototype of a new robot is built.

Concluding, even in this constrained context, real-world partitioners were able to identify some areas for which further studies and testing are required. This includes topics such as risk management and work authorization for robots, day-to-day planning and scheduling, and other tasks where robots introduce a new view of business execution, as robots typically act without deviating from their original plan, while humans frequently adapt their behavior based on context.

3.8 | From documenting the design process and its solution to digital twins

PRODEC can be considered an incremental documentation process of an agile engineering design process, and its solutions are progressively made tangible. It is, therefore, based on the idea that design is supported by clear documentation of what should be and has been done during the life cycle of an STS. Traditional technical and operational documentation support has been paper-based until recently. Both technical and operational documents shifted progressively to digital media for the last 20 years. However, there are still additional independent support systems. The upcoming digital engineering combined with our human-centered approach is changing the perspective. Developing virtual human-centered prototypes becomes a way to develop interactive digital documents that not only represent the systems being developed but also offer potential human-in-the-loop simulation support. Moreover, these new capabilities can be described as digital twins (DTs) of the systems under development and operations.

More specifically, HITLS is currently evolving toward the digital twin (DT) concept and tools. A DT is a virtual instance of a physical/cognitive system that simulates dynamic phenomena of both structures and functions of a system. The DT concept was presented to the industry in 2002 under the term "Conceptual Ideal for Product Lifecycle Management (PLM)" at the University of Michigan.[79] NASA adopted the term "digital twin."[80–83] A Digital Twin is "a set of virtual information constructs that fully describes a potential or actual physical manufactured product from the micro atomic level to the macro geometrical level" (Grieves, 2016). This definition usually refers to a product's structure. It should be extended to the product's functions. For AIAA, a digital twin (DT) is "a set of virtual information constructs that mimics the structure, context, and behavior of an individual/unique physical asset, or a group of physical assets, is dynamically updated with data from its physical twin throughout its life cycle and informs decisions that realize value."[84(p. 5)] In this article, a DT mimics a physical and conceptual entity. DTs have been successfully used in many application domains and are considered an important aspect of Model-Based Systems Engineering (MBSE).[85] In this article, a DT can also support HITLS and model-based HSI.[86] Figure 17 presents a concept map of a digital twin.

Madni et al.[87] claimed that DT extends MBSE and can be used as a model of a real-world system to represent and simulate its structure, performance (i.e., function), health status, and mission-specific characteristics during the whole life cycle of the system and incrementally update it from experience (e.g., malfunctions experienced, maintenance, and repair history). In other words, a digital twin can receive experience feedback information and support for system performance (e.g., preventive and timely maintenance based on knowledge of the

FIGURE 17 Digital twin definitions and properties.

system's maintenance history and observed system behavior). A digital twin is, therefore, a great support to improve understanding of the various relationships between system design and usage. In addition, a digital twin enables to support traceability and logistics along the whole life cycle of a system.

Considering a digital twin as a system digital model, in the SE sense, a distinction should be made between predictive DT and explanatory DT. A predictive DT is typically a well-tested digital analog that produces similar outputs as the system would produce in response to the same inputs. It is usually simple and defined in a limited context. It is consequently short-term, rigid, and focused on a specific process or phenomenon. It can be used for marketing and sometimes when the domain is sufficiently mastered to predict crucial system states operationally.

In contrast, an explanatory DT is based on a taxonomy of the domain incrementally developed. It is longer-term, flexible, and generic within the domain being considered. It can be used to analyze, design, and evaluate a complex system and to document its design and development process and its evolutionary solutions. Therefore, a digital twin is a digital model that enables running simulations to predict the behavior and performance of a real-world system and/or explain why this system behaves the way it behaves.

A digital twin could be considered as a sophisticated interactive notebook that provides a vivid representation of the system being considered. Validation and certification of a DT is never finished. A DT is constantly modified by integrating new features

and modifying and/or removing old ones. Considering the system's life cycle, digital twins can be used in a variety of industrial activities: product design, engineering optimization, smart manufacturing, job-shop, scheduling, human–machine collaboration, operations diagnostic and decision-making, prognostics and health management, maintenance management; and life-cycled product data management.

A DT could be used as a mediating tool to collaboratively test a very early concept within a design team that includes a group of experts with different backgrounds. Note that the design team includes targeted users or human operators of the system to be developed. Each design team member should understand the same thing as the others. This is why team members should have the same objectives and share the same situation awareness (SA) of what is being designed and further developed.[88] In this case, the DT represents this shared SA (SSA). Each design team member can see the same thing and eventually manipulate it. It is, therefore, a great support for participatory design. A DT for SSA starts with a Discount DT (DDT). This could be done with the help of an artist capable of producing a DT in the form of a cartoon or animation of the targeted system to be developed. DDT increases design team inter-subjectivity through incremental modifications using collective critical thinking and experience feedback.

Once the DT is fully completed and approved by all design team members, it can be used to develop Computer Aided Design (CAD) of the system in depth. Using real numbers to determine the system's structure and function makes the whole thing tangible. The DT then

becomes more rational and tangible. Once, a "final" version is approved, physical construction of the system can start. This is another stage of tangibilization. Once a satisfactory version of the physical prototype is constructed, it can be tested in the "real world." Nevertheless, handling this HCD participatory process toward HSI requires everyone to speak the same systemic language. Consequently, the system concept should be clarified to include the human operators who complete functions necessary for system operation within the deployed environment as part of the system.

4 | DISCUSSIONS

4.1 | Comparison to other related work

The distinction between procedural and declarative (or logical) knowledge is familiar.[89] This distinction was used to denote procedural and declarative programming in the early stages of AI. Procedural programming languages are high-level abstractions of computer instructions that enable the programmer to express an algorithm in a line-by-line sequence of instructions. Procedural programming languages originated from FORTRAN,[90,91] and include Pascal,[92] C,[93] and Python.[94] Conversely, declarative programming languages enable programmers to declare a set of objects that have properties and capabilities. They originate from LISP[95] and include Haskell, Caml, and SQL. These objects are further processed as they are by a software inference engine. Object-oriented programming originated from Smalltalk and was inherited from both paradigms, including Java and C++. These object-oriented languages enable programmers to declare objects and their properties and object-specific procedures called methods.

Computer science and cognitive psychology cross-fertilized for a long time. Indeed, procedural knowledge and its distinction with declarative (or conceptual) knowledge has been developed in several fields related to cognition, such as educational science[96] and development psychology,[97] including mathematics education,[98–100] user modeling,[101] experimental psychology.[102–104]

Czachorowski et al.,[105] focused on a company's offshore oil and gas supply chain operations to identify areas for improvement. They developed AS-IS and TO-BE scenarios to understand better which directions to take to support current operations and the potential to change certain aspects. However, they would have benefited from remote mobile robotics as we did. Another major difference in their work is implementing field HITLS and iterative formative evaluations based on activity analysis and emerging function identification based on empirical experience.

Back to the theater metaphor, a theatrical play is usually available first in a procedural manner. A writer produces an essay that tells a story. Then, a producer selects, in a declarative manner, actors who must read the essay and learn their roles and scripts procedurally and coordinates them. PRODEC has been designed to be used in HCD to benefit from both operations experience (i.e., human operators will be asked to provide their salient operations stories) and definition of objects and agents involved in the targeted human–machine system

to be designed (i.e., the design team will provide prototypes at various progressive levels of maturity).

With this method, we first explore how operations are performed prior to starting any design. A procedural scenario is developed with experienced people, as a timeline of events. PRODEC is based on the claim that stories told by subject matter experts can be easily translated into procedural scenarios. Once one or several procedural scenarios are elicited, human-centered designers can extract meaningful objects and agents that are functionally and structurally described. This constitutes declarative scenarios (i.e., organizational configurations). Of course, this articulation of procedural and declarative knowledge can and should be repeated as many times as necessary and possible to get a consistent and implementable human–machine system prototype, which will be further developed and validated.

The production of procedural and declarative knowledge should be guided by a framework that supports the main factors that include artifacts to be designed and developed, users who will use them, the various tasks to be executed by these artifacts, the organizational environment where they will be deployed, and the various critical situations in which these tasks will be executed.

4.2 | Opening a discussion on contemporary human systems integration

In this article, we have chosen a specific project to illustrate the genesis and use of PRODEC. Other articles are being written to present the problems and potential solutions other PRODEC-supported projects. The following question must be addressed: to what extent is the analysis using PRODEC subjective? Indeed, science requires us to demonstrate that different experts (operators or engineers) who analyze the same system arrive at the same PRODEC solution. Therefore, it is legitimate to address the issue of the influence of subjective judgment in PRODEC. As already said in the introduction, engineering design is about domain expertise and experience (e.g., aerospace, oil and gas, medicine), which can lead to many subjective judgments when human experts are involved, and creativity is required to find possible feasible solutions, which is subjective. This discussion brings back the problem of subjectivity in science and engineering. Whenever we need to make sense of something, we provide an explanation that necessarily comes from a human being. Therefore, sense-making is a subjective process of situation awareness and appropriation. How can we increase the credibility of a proposed solution? When several independent experts positively evaluate a solution using their experience (i.e., subjective judgment), the credibility in that solution necessarily increases. How many experts must we decide if the solution is credible/acceptable? We adopt the following peer review heuristic, as currently practiced in evaluations of research contributions: three expert reviewers per solution. Metrics are generally defined for these evaluations, and the contextual quality[‡‡] of expert reviewers is

---

‡‡ Contextual quality is expressed as the reviewer's expertise appropriate for evaluating the solution.

crucial. Another question is: how does PRODEC differentiate from traditional approaches to task analysis during design and development? Task analysis has been performed for a long time, usually before the design process begins, but not incrementally during the design process, by augmenting the initial task models with the discovered emergent activities. What we used to call task models can be generalized to the living memory of incrementally informed HCD activity models during the life cycle of a system. PRODEC provides a language (i.e., a knowledge representation) that supports system design where the following statement applies: "writing as design and design as writing." The concept of "writing" must be extended to multi-media, or even hypermedia, development (i.e., writing on paper is replaced by the production of a multi-media documentation system). In fact, this multimedia documentation system is used for the human-in-the-loop simulation, which is nothing else than a digital twin of the system under conception, development, and incrementally transformation.

5 | CONCLUSION AND PERSPECTIVES

At this point, can we define PRODEC? PRODEC is an SBD method that supports HSI of a socio-technical system (STS) by continuously capturing appropriate contextualized procedural scenarios from subject matter experts and implementing declarative configurations in the form of digital twins that support human-in-the-loop simulations (HITLSs), which enable incremental discovery of emergent properties of the STS. PRODEC is, therefore, an iterative HSI approach. Digital twins are used as prototypes of physically tangible system prototypes and as figuratively tangible signifiers of the systems they represent. From the point of view of model-based systems engineering, they are digitally implemented models that can be used in HITLS.

PRODEC breaks with the conventional approaches based on task analysis and user interface (UI) design. Task analysis is normative and less effective than activity analysis can be. In addition, UI design usually occurs once a complex system is fully developed to adapt human operators to the machines almost fully developed. In contrast, PRODEC supports an SBD approach that allows, from the early stages of the design process, the involvement of potential future human operators who participate with both creative stakeholders and experienced experts in the development and test of virtual prototypes, incrementally made tangible in an agile way. User interfaces are considered as components of the overall system from the beginning of the design process. PRODEC proposes models and notations to handle this human- and organization-centered design process that supports eliciting emergent functions and structures. The main difference between PRODEC's HSI approach and most technology-centric systems engineering approaches lies in the imperative articulation of a context architecture with a system architecture (Figure 18).

Indeed, just as a system can be seen as a SoS, a context can also be represented as a context of contexts. Technical designers typically construct systems making declarative configurations explicit, and contexts are obtained from operations experts from formalizing pro-
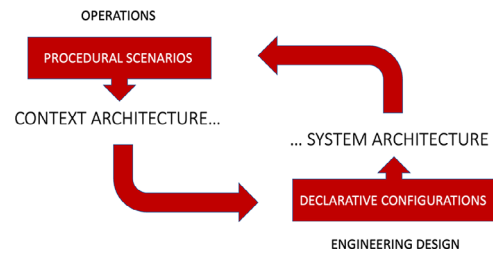

FIGURE 18   Context and system architectures.

cedural scenarios. PRODEC is a formal method that links procedural scenarios of tasks and activities (AS-IS and TO-BE) in the context space to declarative configurations in the system space. It is typically important to structure the context architecture using various kinds of categorizations, including context patterns related to time, space, and other dimensions such as, intrinsic and extrinsic patterns contextually relevant within the overall system of systems being considered.

In addition to the oil and gas tele-robotics sector partially described in this article, PRODEC is currently used in other projects dealing with aerospace[17] and healthcare sector.[106] This cumulative PRODEC experience shows that it can be useful in a wider variety of HSI projects that require knowledge acquisition and modeling of complex human–machine multi-agent systems, especially life-critical systems. A good question is, for example, what levels of granularity matter for assessing human operator's trust in increasingly autonomous systems such as robots? What kind of metrics should be developed? How could we measure the maturity of the human–machine system being developed? To answer these questions, this article provides a methodology and conceptual tools that would be useless without subject matter experts (i.e., experienced people), good scenarios, and appropriate metrics and criteria.

Finally, PRODEC proposes a set of metrics dealing with cognition, life-cycled system evolution, and maturity. More specifically, our current focus in the multi-scale maturity evaluation process that supports TOP model-based formative evaluations (i.e., enables both testing and revising technology, organizations, and people's competencies[§§] requirements concurrently). Answers to these questions will develop through iterative uses of PRODEC SBD approach combined with human-in-the-loop simulations based on testable and usable digital prototypes progressively made tangible.

---

§§ Competencies include skills, rules, and knowledge in Rasmussen's terms.[73]

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available on request from the corresponding author.

ORCID

Guy André Boy https://orcid.org/0000-0002-0497-7949

REFERENCES

1. INCOSE. INCOSE Systems Engineering Handbook, 5th Edition, 2023. ISBN-13 978-1119814290.
2. Rosson MB, Carroll JM. Scenario-based design. In: Jacko J, Sears A, eds. *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications.* Lawrence Erlbaum Associates; 2002:1032-1050.
3. Rosenberg N. Learning by using. In: *Inside the Black Box: Technology and Economics.* 1st ed. Cambridge University Press; 1982:120-140.
4. Johnson M, Bradshaw JM, Feltovich PJ, Jonker CM, Van Riemsdijk MB, Sierhuis M. Coactive design: designing support for interdependence in joint activity. *J Hum Robot Interact.* 2014;3(1):43. doi:10.5898/JHRI.3.1.Johnson
5. Boy GA. *Tangible Interactive Systems.* Springer; 2016.
6. Demeure A, Masson D, Calvary G. Graphs of models for exploring design spaces in the engineering of human-computer interaction. Paper presented at: *IUI 2011 - International Conference on Intelligent User Interfaces;* Feb 2011; Palo Alto, CA, USA.
7. Hackos J, Redish J. *User and Task Analysis for Interface Design.* Wiley; 1998.
8. Crystal A, Ellington B. Task Analysis and Human-Computer Interaction: Approaches, Techniques, and Levels of Analysis: *Proceedings of the Tenth Americas Conference on Information Systems,* New York, August; 2004.
9. Emberey CL, Milton NR, Berends JPTJ, Van Tooren MJL, Van der Elst SWG, Vermeulen B. Application of knowledge engineering methodologies to support engineering design application development in aerospace. In: *Proceedings of the 7th AIAA ATIO Conference, 2nd CEIAT International Conference on Innovation and Integration in Aero Sciences, 17th LTA Systems Tech Conf; followed by 2nd TEOS Forum;* 2007: 7708.
10. Ahmed S, Hacker P, Wallace K. The role of knowledge and experience in engineering design. In: *DS 35: Proceedings ICED 05, the 15th International Conference on Engineering Design,* Melbourne, Australia; 2005: 206-207.
11. Van der Elst SWG, Van Tooren MJL. Application of a knowledge engineering process to support engineering design application development. In: Curran R, Chou S-Y, Trappey A, eds. *Collaborative Product and Service Life Cycle Management for a Sustainable World.* Springer; 2008:417-431.
12. Lang GT, Eberts RE, Gabel MG, Barash MM. Extracting and using procedural knowledge in a CAD task. *IEEE Trans. Eng. Manage.* 1991;38(3):257-268.
13. Ertas A, Jones J. *The Engineering Design Process.* 2nd ed. John Wiley & Sons, Inc.; 1996.
14. Dym CL, Little P. *Engineering Design.* 3rd ed. John Wiley & Sons, Inc; 2009.
15. Lawrence PR. How to deal with resistance to change. *Harv. Bus. Rev.* 1969. https://hbr.org/1969/01/how-to-deal-with-resistance-to-change. Retrieved on March 1, 2024.
16. Norman DA. *Design of Everyday Things.* The MIT Press; 2013. Revised and expanded edition. ISBN-13: 978–0262525671.
17. Boy GA, Morel C. The machine as a partner: human-machine teaming design using the PRODEC method. *Work.* 2022;73(s1):S15-S30. 10.3233/WOR-220268
18. Boehm-Davis DA, Durso FT, Lee JD. *APA Handbook of Human Systems Integration.* 1st ed. American Psychological Association; 2015. ISBN-13: 978-1433818288.
19. Boy GA, Kennedy GE. *INCOSE Human Systems Integration Primer Volume 1.* International Council on Systems Engineering; 2023.
20. Bellet T, Bornard JC, Mayenobe P, Paris JC, Gruyer D, Claverie B. A virtual human centered design tool based on a cognitive simulation model of the car driver. Paper presented at: *European Conference on Human Centered Design for Intelligent Transport Systems;* 2012; Valence, Spain.
21. Karwowski W. *International Encyclopedia of Ergonomics and Human Factors.* CRC Press; 2006. ISBN 0-415-30430-X 9780415304306.
22. Sollenberger RL, Willems B, Della Rocco PS, Koros A, Truitt T. *Human-in-the-Loop Simulation Evaluating the Collocation of the User Request Evaluation Tool, Traffic Management Advisor, and Controller- Pilot Data Link Communications: Experiment I—Tool Combinations.* National Technical Information Service; 2005. Technical Report DOT/FAA/CT-TN04/28.
23. Minsky M. The Society of Mind. Simon & Schuster; 1985.ISBN 0-671-60740-5.
24. Boy GA. Cross-fertilization of human-systems integration and artificial intelligence: looking for systemic flexibility. Paper presented at: *Proceedings of AI4SE 2019, First Workshop on the Application of Artificial Intelligence for Systems Engineering;* 2019; Universidad Carlos III de Madrid Leganés, Madrid, Spain.
25. Miller ME, Mcguirl JM, Schneider MF, Ford TC. Systems modeling language extension to support modeling of human-agent teams. *Syst Eng J.* 2020;23:519-533. https://onlinelibrary.wiley.com/doi/abs/10.1002/sys.21546. Published online.
26. Boy GA. *Human Systems Integration: From Virtual to Tangible.* CRC Taylor & Francis Press; 2020a.
27. Diederich J, Ruhmann I, May M. KRITON: a knowledge-acquisition tool for expert systems. *Int J Man Mach Stud.* 1987;26(1):29-40.
28. Sobolewsk M. Multiagent knowledge-based environment for concurrent engineering applications. *Concurr Eng.* 1996;4(1):89-97.
29. Musen MA, Fagan LM, Shortliffe EH. Graphical specification of procedural knowledge for an expert system: The User Interface. In: Hendler J, ed. *Expert Syst.* 1988;15-35.
30. Allard JR, Cochran EL, Wolff AS. *Method of construction of hierarchically organized procedural node information structure including a method for extracting procedural knowledge from an expert, and procedural node information structure constructed thereby.* US Patent 5, pp. 471,560. November 28, 1995, pp. 471-560.
31. Martinie C, Palanque P, Ragosta M, Fahssi R. Extending procedural task models by systematic explicit integration of objects, knowledge, and information. In: *Proceedings of the 31st European Conference on Cognitive Ergonomics;* 2013: 1-10.
32. Kasser J, Frank M, Zhao YY. Assessing the competencies of systems engineers. In: *Proceedings of the 7th Bi-Annual European SE Conference (EUSEC), Vol. 26, May 23, 2010,* Stockholm, Sweden; 2010: 2010.
33. Johnson SB. Three approaches to big technology: operations research, systems engineering, and project management. *Technol Cult.* 1997;38(4):891-919.
34. Tolk A, Diallo SY, Turnitsa CD. Applying the levels of conceptual interoperability model in support of integrability, interoperability, and composability for system-of-systems engineering. *J Syst Cybernetics Informat.* 2007;5(5):65-74.
35. Frank M, Kasser J. Assessing the capacity for engineering systems thinking (CEST) and other competencies of systems engineers. In: Cogan B, ed. *Systems Engineering-Practice and Theory.* INTECH Open Access Publisher; 2012:217-230.
36. Boy GA. *Design for Flexibility.* Springer Nature; 2021. ISBN 3030763919, 9783030763.

37. Madni AM, Madni CC. Sievers M. Adaptive Cyber-Physical-Human Systems. *Proceedings of the 2018 INCOSE International Symposium*, Washington, DC, USA, 7-12 July 2018.

38. Madni AM, Orellana DW. Extending model-based systems engineering to address human-systems integration considerations in the system life cycle. *Sys Con.* 2018:1-7.

39. Orellana DW, Madni AM. Human system integration ontology: enhancing model-based systems engineering to evaluate human-system performance. *CSER.* 2014:19-25.

40. Raymond C, Prun D. Extending MBSE Methodology and SysML Formalism to Integrate Human Considerations. HCI-Aero 2016, International Conference on Human-Computer Interaction in Aerospace; September 2016; Paris, France.

41. Watson ME, Rusnock CF, Colombi JM, Miller ME. Human-centered design using system modeling language. *J Cogn Eng Decis Mak.* 2017;11(3):252-269. doi.org/10.1177/1555343417705255

42. Boy GA. Knowledge management for product maturity. Paper presented at: Proceedings of the International Conference on Knowledge Capture (K-Cap'05); October 2005; Banff, Canada. Also, in the ACM Digital Library.

43. Igbinovia MO, Ikenwe IJ. Knowledge management: processes and systems. *J Inf Knowl Manag.* 2018;8(3):26. doi:10.4314/iijikm.v8i3.3

44. Leplat J. *Repères pour l'analyse de l'activité en ergonomie [Benchmarks for Activity Analysis in Ergonomics].* Presses Universitaires de France, Paris; 2009.

45. Stanton NA, Salmon PM, Rafferty LA, Walker GH, Baber C, Jenkins DP. *Human Factors Methods: A Practical Guide for Engineering and Design.* 2nd ed.; 2013. doi:10.1201/9781315587394

46. Salas E, Maurino D. *Human Factors in Aviation.* 2nd ed. Academic Press; 2010. ISBN-13: 978–012374518.

47. Arnold JE. Creative Engineering: Promoting Innovation by Thinking Differently. Edited with an Introduction and Biographical Essay by William J. Clancey. *Stanford Digital Repository*; 1959. https://stacks.stanford.edu/file/druid:jb100vs5745/Creative%20Engineering%20-%20John%20E.%20Arnold.pdf Retrieved 15 November 2023

48. Boy GA, Narkevicius J. Unifying human centered design and systems engineering for human systems integration. In: Aiguier M, Boulanger F, Krob D, Marchal C, eds. *Complex Systems Design and Management.* Springer; 2013. 2014. ISBN-13: 978-3-319-02811-8.

49. Plimmer B, Apperley M. Freeform: a tool for sketching form designs. In: BCS HCI Conference, Vol. Proceedings of the BCS HCI Conference. British HCI Group; 2003: 183-186.

50. Friedenthal S, Moore A, Steiner R. *A Practical Guide to SysML.* Elsevier Science & Technology; 2014.

51. Delligatti L. *SysML Distilled: A Brief Guide to the Systems Modeling Language.* Addison-Wesley; 2013.

52. Jarraya Y, Debbabi M, Bentahar J. On the meaning of SysML activity diagrams. Paper presented at: Conference: 16th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems, ECBS 2009; April 14–16, 2009; San Francisco, CA, USA. doi:10.1109/ECBS.2009.25

53. White SA. Business Process Modeling Notation; 2004. Accessed November 23. https://web.archive.org/web/20130818123649/http://www.omg.org/bpmn/Documents/BPMN_V1-0_May_3_2004.pdf

54. White SA, Bock C. BPMN 2.0 Handbook Second Edition: Methods, Concepts, Case Studies, and Standards in Business Process Management Notation. *Future Strategies Inc.*; 2011. ISBN 978-0-9849764-0-9.

55. Boy GA. *Cognitive function analysis. In: Contemporary Studies in Cognitive Science and Technology Series.* Ablex-Praeger Press; 1998. https://www.amazon.com/Cognitive-Function-Analysis-Contemporary-Technology/dp/1567503772

56. Cox A, Szajnfarber Z. System user pathways to change. In: Madni AM, Boehm B, Wrwin DA, Ghanem R, and Wheaton MJ, eds. *15th Annual Conference on Systems Engineering Research Disciplinary Convergence: Implications for Systems Engineering Research*; 2017: 1–9.

57. Gavrilova T, Andreeva T. Knowledge elicitation techniques in a knowledge management context. *J Knowl Manag.* 2012;16(4):523-537. doi:10.1108/13673271211246112. Emerald Group Publishing Limited. ISSN 1367–3270.

58. Hoffman RR, Shadbolt NR, Burton AM, Klein G. Eliciting knowledge from experts: a methodological analysis. *Organ Behav Hum Decis.* 1995;62(2):129-158.

59. Boose JH. A survey of knowledge acquisition techniques and tools. *Knowledge Acquisition.* 1989;1(1):3-37.

60. Doerr HO, Carr JE. Videotape 'self-confrontation effect' as function of person viewed, and task performed. *Percept. Mot. Skills.* 1982;54(2):419-433.

61. Mollo V, Falzon P. Auto-confrontation and allo-confrontation as tools for reflective activities. *Appl Ergon.* 2004;35(6):531-540.

62. Muller MJ. Participatory design: the third space in HCI (revised). In: Jacko J and Sears A, eds. *Handbook of HCI.* 2nd ed. Erlbaum, Mahway; 2007.

63. Plattner H, Meinel C, Larry Leifer L. Design Thinking Research—Taking Breakthrough Innovation Home. *Understanding Innovation Series.* Springer International Publishing Switzerland; 2016.

64. Georgeff MP, Lansky AL. Procedural knowledge. *Proc. IEEE.* 1986;74(10):1383-1398.

65. Fernandez J, Bhat J. Addressing the complexities of global process harmonization. In: Wang M, Sun Z, eds. *The Handbook of Research on Complex Dynamic Process Management: Techniques for Adaptability in Turbulent Environments.* Information Resources Management Association; 2010. doi:10.4018/978-1-60566-669-3.ch015

66. Spaulding E, Faste H. Design-driven narrative: using stories to prototype build immersive design worlds. In: *Proceedings of ACM CHI'13 Conference: Changing Perspectives*, Paris, France; 2013.

67. Endsley MR. Toward a theory of situation awareness in dynamic systems. *Hum Factors.* 1995;37(1):32-64. doi:10.1518/001872095779049543

68. Wickens CD. Effort in human factors performance and decision making. *Hum Fact.* 2014;56(8):1329-1336. doi:10.1177/0018720814558419

69. Boy GA, Brachet G. *Risk Taking: A Human Necessity that Needs to be Managed.* Air and Space Academy; 2010. https://academieairespace.com/en/publications-2/?_page=3

70. Wickens CD. Multiple resources and mental workload. *Hum Factors.* 2008;50(3):449-455. doi:10.1518/001872008x288394

71. Salazar G, See JE, Handley HAH, Craft H. Understanding human readiness levels. Proc Human Fact Ergon Soc Annual Meet 2020;64(1):1765-1769. doi:10.1177/1071181320641427

72. Munoz JM, Bartoli E, Macrez S. Operation room: a key component for upscaling robotic solutions on site. Communication presented at the Abu Dhabi International Petroleum Exhibition and Conference (ADIPEC), Abu Dhabi; 2020. doi:10.2118/202667-MS

73. Rasmussen J. *Information Processing and Human-Machine Interaction.* Elsevier; 1986.

74. Salber D, Coutaz J. Applying the Wizard of Oz technique to the study of multimodal systems. In: Human-Computer Interaction, Third International Conference, EWHCI '93, Moscow, Russia; 1993. doi:10.1007/3-540-57433-6_51

75. Taylor RM. Situational Awareness Rating Technique (SART): the development of a tool for aircrew systems design (AGARD-CP-478) pp3/1 –3/17. In: *Situational Awareness in Aerospace Operations.* NATO-AGARD; 1990.

76. Hart SG, Staveland LE. Development of NASA-TLX (Task Load Index): results of empirical and theoretical research. In: Hancock PA and

Meshkati N, eds. *Human Mental Workload*. North Holland Press; 1988.

77. Sellers JM. *Team Workload Questionnaire (TWLQ): Development and Assessment of a Subjective Measure of Team Workload*. Psychology. Dissertation. University of Canterbury; 2013. doi.org/10.26021/8145

78. Brooke J. SUS—a quick and dirty usability scale. In: Jordan PW, Thomas B, Weerdmeester BA, McClelland I, eds. *Usability Evaluation in Industry*. Taylor & Francis; 1996:189-194.

79. Grieves M. Origins of the digital twin concept. Florida Institute of Technology Report. 2016. doi:10.13140/RG.2.2.26367.61609

80. Caruso P, Dumbacher D, Grieves M. Product lifecycle management and the quest for sustainable space explorations. Paper presented at: *AIAA SPACE 2010 Conference and Exposition*, Anaheim, CA; 2010.

81. Glaessgen EH, Stargel D. The digital twin paradigm for future NASA and US Air Force vehicles. Paper presented at: *AAIA 53rd Structures, Structural Dynamics, and Materials Conference*, Honolulu, Hawaii; 2012.

82. Piascik R, Vickers J, Lowry D, Scotti S, Stewart J, Calomino A. Technology Area 12: Materials, Structures, Mechanical Systems, and Manufacturing Road Map. *NASA Office of Chief Technologist*; 2010.

83. Tuegel EJ, Ingraffea AR, Eason TG, Spottswood SM. Reengineering aircraft structural life prediction using a digital twin. *Int J Aerosp Eng*. 2011:154798.

84. AIAA. *Digital Twin: Reference model, realizations, and recommendations. Authored by the American Institute of Aeronautics and Astronautics (AIAA) Digital Engineering Integration Committee*. Approved by the AIAA Public Policy Committee; 2023.

85. Bordeleau F, Combemale B, Eramo R, van den Brand M, Wimmer M. Towards model-driven digital twin engineering: current opportunities and future challenges. In: *Systems Modelling and Management: First International Conference. ICSMM* Springer; 2020.

86. Boy GA. Model-based human systems integration. In: Madni AM, Augustine N, eds. *In the Handbook of Model-Based Systems Engineering*. Springer; 2023. doi:10.1007/978-3-030-27486-3_28-1

87. Madni A, Madni C, Lucero S. Leveraging digital twin technology in model-based systems engineering. *Systems*. 2019;7:7. doi:10.3390/systems7010007

88. Endsley MR, Jones DG. *Designing for Situation Awareness: An Approach to User-Centered Design*. 2nd ed. CRC Press; 2011. ISBN 978–1420063554.

89. Cauley KM. Studying knowledge acquisition: distinctions among procedural, conceptual and logical knowledge. Paper presented at: *Proceedings of the 67th Annual Meeting of the American Educational Research Association conference*; April 16–20, 1986; San Francisco, CA.

90. McCracken DD. *A Guide to FORTRAN Programming*. Wiley; 1961. LCCN 61016618.

91. Kupferschmid M. *Classical Fortran: Programming for Engineering and Scientific Applications*. CRC Press; 2002. ISBN 978-0-8247-0802-3.

92. Wirth N. The Programming Language Pascal. *Acta Informatica*. 1971;1:35-63.

93. Prinz P, Crawford T. *C in a Nutshell: The Definitive Reference*. 2nd ed.; 2015. Kindle Edition. B0197CH96O.

94. Deitel PJ, Deitel H. *Python for Programmers: with Big Data and Artificial Intelligence Case Studies*. Pearson Higher Ed; 2019. ISBN-13: 978-0135224335.

95. Berkeley EC, Bobrow DG, eds. *The Programming Language LISP: Its Operation and Applications*. MIT Press; 1964.

96. Mccormick R. Conceptual and procedural knowledge. *Int J Technol Des Educ*. 1997;7(1-2):141-159.

97. Schneider M, Rittle-Johnson B, Star JR. Relations among conceptual knowledge, procedural knowledge, and procedural flexibility in two samples differing in prior knowledge. *Dev Psychol*. 2011;47(6):1525.

98. Star JR. Reconceptualizing procedural knowledge. *Math Educ Res J*. 2005:404-411.

99. Hiebert J, Lefevre P. Conceptual and procedural knowledge in mathematics: an introductory analysis. In: *Conceptual and Procedural Knowledge: The Case of Mathematics*; 1986:1-27.

100. Carpenter TP. Conceptual knowledge as a foundation for procedural knowledge. In: Hiebert J, ed. *Conceptual and Procedural Knowledge: The Case of Mathematics*. Lawrence Erlbaum Associates; 1986:113-132.

101. Corbett AT, Anderson JR. Knowledge tracing: modeling the acquisition of procedural knowledge. *User Model User-Adapt*. 1994;4(4):253-278.

102. Willingham DB, Nissen MJ, Bullemer P. On the development of procedural knowledge. *J Exp Psychol Learn Mem Cogn*. 1989;15(6):1047.

103. Lewicki P, Czyzewska M, Hoffman H. Unconscious acquisition of complex procedural knowledge. *J Exp Psychol Learn Mem Cogn*. 1987;13(4):523.

104. Richard JF. *Logique de fonctionnement et logique d'utilisation*. Rapport de recherche no. 202, INRIA; 1983.

105. Czachorowski KV, Haskins C, Mansouri M. Minding the gap between the front and back offices: a systemic analysis of the offshore oil and gas upstream supply chain for framing digital transformation. *Syst Eng*. 2023;26:241-256. https://incose.onlinelibrary.wiley.com/doi/10.1002/sys.21652

106. Touzet M, Boy GA. *INNOMED Project Work in Progress on the Reorganization of General Practice Medicine in France*. ESTIA Institute of Technology; 2021.