OSCAR: Operational System Context Acquisition and Representation

Alexandre Disdier *EstiaR*
Univ. Bordeaux, ESTIA-Institute of Technology
F-64210 Bidart, France
alexandre.disdier@estia.fr

Dimitri Masson *EstiaR*
Univ. Bordeaux, ESTIA-Institute of Technology
F-64210 Bidart, France
d.masson@estia.fr

March 31, 2026

Abstract

Sociotechnical systems are characterized by complex and emergent behaviors shaped by their interactions with the context in which they operate. Despite advances in Systems Engineering and Human Systems Integration, the concept of operational context remains underdeveloped and insufficiently integrated into modeling practices. This gap hinders the ability of designers to capture contextual dynamics of a system during early-stage design processes. To address this, we propose the OSCAR (Operational System Context Acquisition and Representation) framework, which formalizes the relationship between the four spaces inherent to complex sociotechnical systems: functions, structures, environment, and their evolution over time. By distinguishing and connecting these four spaces, OSCAR enables the systematic representation of how contextual elements interact with a system of interest during operations and how they evolve towards new situations. We illustrate applicability of the proposed framework through an air traffic control example and discuss how it could be harnessed within practical approaches to sociotechnical system design.

Keywords: human systems integration, context, system modeling, sociotechnical systems

1 Introduction

Complex Sociotechnical Systems (STSs) display emergent properties that are hard to anticipate at design time. A major driver emergence is the system's sensitivity to its operational context [1]. Recent advances in Systems Engineering (SE) and Human Systems Integration (HSI) [2] have emphasized the need for rigorous modeling and simulation to acquire essential knowledge about systems before they become operational. A central objective of HSI-driven design is to capture and understand emergent behaviors of STSs as early as possible in the lifecycle, when design flexibility is still high and corrective actions can be implemented at lower cost [3].

One persistent challenge in STS design, from a HSI perspective, lies in defining and abstracting the system of interest from its operational environment [4]. STSs operate within an inherently complex and dynamic *context* that encompasses technological, organizational, and human entities [3]. This context is itself system-like, interacting continuously with the STS and influencing its behavior and performance. To our best knowledge in SE and system design, there remains no unified framework capable of identifying and integrating both system-level and context-level properties within complex STSs.

Furthermore, existing systems modeling approaches provide limited mechanisms to represent this operational context. Common modeling languages such as SysML, UML, BPMN, and Object-Process Methodology (OPM) lack the expressiveness required to capture contextual dependencies, interactions, and evolution within engineering models. As a result, context modeling is often omitted or introduced too late in the design lifecycle, when adaptation costs and design constraints are already high [5].

This limitation reflects a broader challenge in modeling theory: much of the real-world organization surrounding engineered systems resists representation through simple and general models. As noted by Edmonds, "*much of the world around us is organized in a way that is beyond adequate modeling in a sufficiently simple and general manner to cope with*" [6]. Nevertheless, modeling context remains feasible, provided that deliberate trade-offs are made among simplicity, generality, validity, and formality. These trade-offs should be guided by the modeler's objectives and supported by a rigorous understanding of what constitutes context and how it co-evolves with the system of interest.

Recent research has begun to address this gap by proposing an HSI-compatible definition of operational context as "a historical sequence of situations, triggered by events, that influence and help explain the behavior of a focus"—where a *focus* is a couple defined by a structure together with one of its allocated functions [7]. From this definition, six core properties of context have been identified: specific, curated, holistic, transient, entangled, and persistent. These properties emphasize that only a fraction of the world's knowledge is relevant at a time; that relevance depends on the structure–function focus; that context evolves; that system and context co-influence each other; and that past situations can continue to matter.

Other HSI advances such as PRODEC [8] advocate tightly coupling of procedural scenario knowledge with declarative system configurations (i.e., articulating a *context architecture* alongside a *system architecture*), and using human-in-the-loop simulation to reveal emergent behaviors. However, even these approaches lack a compact, design-ready representational scheme that makes context legible and manipulable across multiple levels of abstraction throughout the lifecycle.

This paper introduces OSCAR (Operational System Context Acquisition and Representation), a conceptual framework and modeling viewpoint that integrates context and system representation for STS design. OSCAR organizes context into four coupled *spaces*: (i) a *function* space (roles and resources allocated to achieve goals), (ii) a *structure* space (human, organizational, and technical entities and their interfaces), (iii) an *environment* space (contextual elements external to the system boundary), and (iv) a *time* space (the sequence of situations). By selecting a focus (i.e., a structure and a function from the system of interest), OSCAR projects the subset of contextual elements that are relevant "now," records their evolution as situations in time, and makes explicit how changes in any space propagate to the others. Building on prior HSI/context foundations, OSCAR (1) provides a unified lens to localize and trace context relative to a focus; (2) clarifies mechanisms of situational change via events that alter contextual element values or reconfigure structure–function allocations; and (3) links these changes to resource availability and function feasibility, supporting early trade studies and re-allocation decisions.

This work contributes a conceptual and methodological step forward by (1) formalizing a framework that operationalizes the dynamic properties of context for sociotechnical systems, (2) linking context evolution directly to system structure and function within a unified four-space model, and (3) demonstrating its use for practical HSI-oriented analysis through an air traffic control case study.

Note that this paper is framed as a position paper: the proposed conceptual model synthesizes insights from a broader research program. Our previous work on system context employed multiple qualitative methods, including a systematic literature review on the notion of context in engineering and non-enginering domains [anonymized ref]. The present contribution consolidates these results into a unified conceptual model intended to support a shared understanding of context among systems engineering and, by extension, HSI practitioners.

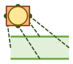
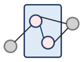
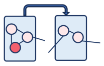
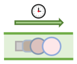
The remainder of this paper is structured as follows. Section 2 gives a brief review of research that has been conducted on system context. Section 3 describes the OSCAR framework and its four spaces in detail. Section 4 illustrates how the OSCAR framework applies to a real-life STS example taken from the air traffic control field. Section 5 provides a discussion on the possible methodological applications of the OSCAR framework for future research. Finally, Section 6 gives concluding remarks.

2 Background research on context

Research on context spans various engineering and non-engineering fields, including linguistics [9], symbolic artificial intelligence [10] and cyber-physical systems [11]. As a result, no single, universally agreed-upon definition of context can be found, and existing definitions are usually very specific to the field in which they have been constructed. In the realm of SE and HSI, the notion of context remains underdeveloped. While the NASA HSI Handbook [2] references the operational or mission context of systems, it does not provide a detailed definition. The SE Handbook from INCOSE [12] and the SE Body of Knowledge [13] describe context as the environment external to the system of interest, but they fail to emphasize how context is characterized and how it evolves at a more granular level. Works from the STSs literature have tried to develop conceptual frameworks of system context grounded to activity theory [14] and human behavior [15], but those are descriptive models that provide limited practical guidance, making it less directly actionable for SE and system design practitioners.

A recent, HSI-compatible definition from Disdier [7] is that context ”*is a historical sequence of situations, triggered by events that influence and help explain the behavior of a focus*”. This definition emphasizes the idea that context is a dynamic, partial representation of the world's knowledge, focusing only on the elements that directly influence the tasks of system agents at any given time. These observations have been translated into the six properties of context listed in Table 1.

Table 1: The six properties of the operational context of a system, taken from [7]

Context property	**Symbology**	**Short description**
Specific		Context is always relative to some focus, defined as a couple (structure; function).
Curated		Only a few contextual elements have a real relevance to the system and its behavior.
Holistic		The whole system's context is more than the individual contexts of its subsystems.
Transient		Context is not static and definitive, but changes through time.
Entangled		Context affects the system's resources, and the system's behavior affects context.
Persistent		A contextual element's former value can still have a relevance to the current situation.

The core idea of these properties is that context is not an absolute entity, but it must be defined relative to a *focus*. A focus is the combination of a specific structure of the STS of interest, together with one of its allocated functions.

Context being curated means that only a small fraction of the world's knowledge is relevant to this focus, i.e., it has an influence whatsoever on how the focus' structure behaves in order to achieve the focus' function.

Although this model established a direct relation between an STS and its context, it remains an high-level view of context, which is defined only at the system-level. The model is limited in the sense that it does not explain how context evolves, how it is characterized and how it interacts with or is influenced by the system at all. Also, it lacks the same practical applicability as other context definitions referenced earlier in this section.

We propose in this paper to supplement this theoretical model and to provide avenues regarding how it could be harnessed in practical STS modeling processes. We present the OSCAR (*Operational System Context Acquisition and Representation*), a conceptual framework that directly relates context and its evolution with the structures and functions that form the STS of interest.

3 Description of the OSCAR framework

In this work, a *system* is understood in the SE sense as an organized set of interacting elements (technical, human, and organizational) structured to perform functions that achieve specific purposes within a defined environment. Each system can therefore be described through two complementary dimensions: its *structure*, representing the configuration and relationships among its constituent elements, and its *function*, representing the activities or transformations those structures perform to fulfill the system's intended objectives. This structural–functional view establishes the foundation upon which context interacts with the system, as the environment influences both the configuration of structures and the performance of functions across different levels of abstraction.

Thus the OSCAR framework consists of four spaces, each contributing to the overall context of the system:

- The function space is the set of functions that are allocated to the structures from the structure space.
- The structure space is a recursive description of the structures that compose the STS of interest.
- the environment space includes the contextual elements that affect and are affected by the system of interest.
- The time space is the sequence of situations succeeding one another in time.

Figure 1 gives a graphical, comprehensive representation of how the four spaces relate to each other. The function, structure and environment spaces respectively define the state of functions, structures and contextual elements at a given time. As stated in [7], in order to define context, one must first choose a focus, i.e., they must first adopt a point of view by choosing a specific structure and a specific function of interest. This idea of a focus is depicted by the camera in the lower right-hand corner of Figure 1. The position of the camera and the direction it is pointing at defines a ray that traverses the function space first, hitting a specific function. When the ray then traverses the structure space, it hits a specific structure. The function and structure that have been hit (i.e., the focus) are projected onto the environment space. In this space, the contextual elements that are close to the projected focus are those that are relevant to it. The set comprising both the focus and the contextual elements that are relevant to it is called a *situation*. Each situation is then printed onto the time space in the background. This space acts like a film reel that records the entire sequence of past and present situations for this particular focus.

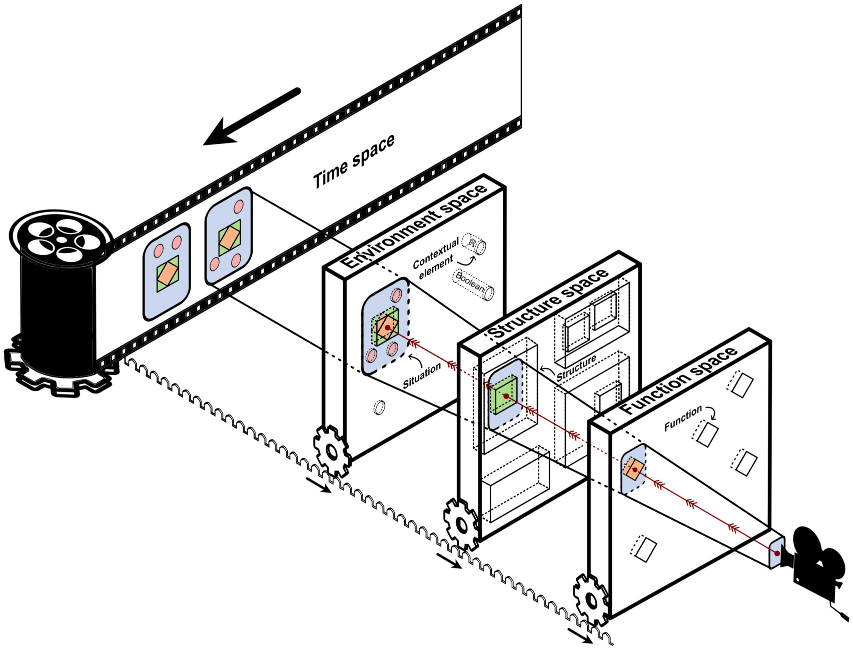


Figure 1: The four spaces of the OSCAR framework

The main strength of this framework is that any change in any space will have an impact on the overall picture, and hence on the final projected situations. For example, if the target structure from the structure space somehow gets moved (e.g. because of an organizational change within the STS), then the line of sight from the camera will eventually hit a different region in the environment space, meaning that a different set of contextual elements will be relevant to the structure's tasks. Likewise, a different functional architecture will lead to different situations.

The framework also encompasses the idea that spaces are not independent from each other. They are all connected to each other by a system of racks and gears at the bottom of the picture. Therefore, when a change occurs in one space, then one or more other spaces may (or may not) be affected. For instance, a change in the function space may affect the structure space (e.g. because the new set of functions might lead to a redesign of the structural architecture of the system and hence a reallocation of function to form). Or, if the structure space changes, the system boundary may need to be redefined, altering the boundary of the environment space at the same time.

The subsequent sections look at each space individually and describe how the space is defined, how it may change and how this change may influence the overall context of the STS.

3.1 The function space

The function space represents the set of functions allocated to the structures that compose the STS of interest. In SE, a function is generally understood as the abstract description of what a system, or one of its elements, is intended to do to achieve specific outcomes [16, 17]. It encompasses the transformations, actions, and activities that convert inputs into desired outputs with defined performance. From a system science perspective, functions express the purposeful behavior of a system's structure, establishing a link between its organization and its goals.

In this work, we adopt a recursive view of systems consistent with the HSI-driven PRODEC framework [8], where each system is composed of structures and functions, both of which can be physical and/or cognitive. Each function has a role, a context of validity, and a set of resources that are themselves systems capable of performing subordinate functions. This recursive relationship allows the emergence of complex, higher-level behaviors from the interactions of lower-level functions and structures, thereby supporting the representation of sociotechnical systems as systems of systems.

Within OSCAR, the function space is central to understanding how a system operates and interacts with its environment, as it defines the purpose and behavior of system structures. Functions are not static constructs; they are instantiated as tasks and activities performed by structures in operational scenarios. The allocation of functions to structures is therefore dynamic, influenced by design decisions, operational constraints, and changes in the environment [18]. Functions serve as the bridge between user needs and their realization through physical, organizational, or cognitive structures [19], forming the basis for system decomposition and integration.

Function allocation must initially be performed with respect to the operating context of the system of interest, ensuring that structures are capable of executing their intended roles under the expected environmental conditions [18]. However, as operational contexts evolve, functions may need to be reallocated among structures to maintain system performance and safety. Such reallocations can propagate through the system, modifying the structure space (through the adaptation or creation of new physical or cognitive entities) and the environment space (by altering the relevance or influence of contextual elements). Consequently, the function space is inherently dynamic, acting as both a driver and a reflection of the continuous co-evolution between system structures and their operational contexts.

3.2 The structure space

The structure space represents the set of entities that compose the STS and their relationships. Structures are the physical, organizational, and cognitive embodiment of what the system is made of. Structures realize system functions. In a STS, structures can be hardware and software components, organizations, or humans. The users of the system are themselves structures in HSI: they are not considered external to the system. This property has been elaborated in recent approaches to system architecture, where humans are explicitly modeled as structural components with their own capabilities, constraints, and cognitive processes [20, 21].

A defining feature of the structure space is its recursive nature: structures are composed of substructures that can themselves be decomposed and organized hierarchically [8]. This assumption makes the structure space comparable to self-similar models such as holonic multi-agent systems [22]. Structures and substructures within the structure space may interact (i.e. change the state of each other [23]) through internal interfaces. Interfaces coordinate flows of matter, energy, and information between structures. Structures within the system boundary interact with enabling systems and other systems from the environment space through external interfaces.

The structural architecture of a STS is not static. Over time, structural evolution arises from technological upgrades, organizational reconfiguration, or the adaptation of human practices. These changes may be planned (e.g. through deliberate system redesign sessions) or emergent (e.g. arising from situated action and improvisation during operations). Such evolution can in turn shift system boundaries, lead to reallocation of functions from the function space, and redefine the nature of both internal and external interfaces, thereby demonstrating the tight coupling of the structure space with the function and environment spaces.

3.3 The environment space

The environment space is the place where all the contextual elements reside. Contextual elements are the formalization of facts about the world. They are projections of the state of the environment of the system. In OSCAR, contextual elements are variables defined on a domain and taking values within a predefined range (e.g. air temperature assigns a real value to a 3D volume around a certain location). The nature of contextual elements may vary significantly depending on the STS of interest. They have different levels of temporality: some contextual elements have persistent values, while other will regularly change their values. They have interdependencies, as the state of one contextual element may affect the state of another contextual element. These interdependencies contribute to emergent behaviors, which can hardly be fully understood by examining individual contextual elements in isolation. Finally, contextual

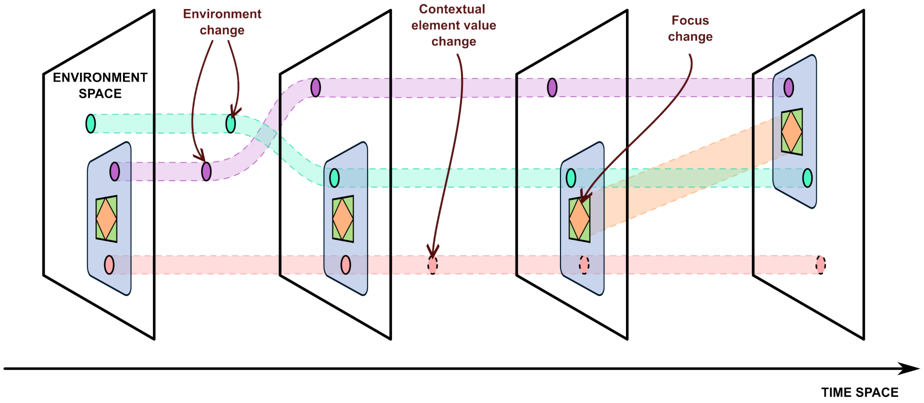


Figure 2: The three factors of situational change

elements may be relevant to a given focus at different levels of criticality, depending on whether the contextual elements will drive system behavior towards nominal, off-nominal or emergency situations.

Once the focus has been set (i.e., a structure and a function of interest have been selected), the ray in Figure 1 will hit the environment space at a certain location and project an area around the hitting point. This area, called a situation, includes the selected focus together with a subset of contextual elements. Those elements within the situation are referred to as "*relevant contextual elements*" because they have an influence on the behavior of the selected structure when performing its selected function. Most contextual elements will lie outside the situation, because only a fraction of the world's knowledge is relevant to the focus at a given time [24].

The contextual elements from the environment space describe the state of any system that interacts with the STS of interest but is not part of it. Such systems can be engineered ones (e.g. enabling systems) or non-engineered ones (e.g. the local atmospheric system). The INCOSE SE Handbook [12] defines the environment of a system as "*the surroundings (natural or man-made) in which the system-of-interest is utilized and supported; or in which the system is being developed, produced or retired*". An important consideration from this definition is that the environment space is not static either but goes through different states between and during the various stages of the system lifecycle. Contextual elements act as dynamic inputs that interact with the STS structures through external interfaces.

Note that the literature review from [7] highlights the fact that many context definitions treat the words "*context*" and "*environment*" as synonyms. In OSCAR, environment is only a partial view of what the context is really about: it contains contextual elements that are external to the system boundaries, whereas context as a whole is the combination of the state of those contextual elements together with the state of the current structures, functions and focus over time.

3.4 The time space

The transient property from Table 1 denotes the dynamic nature of context. This means that a given situation lasts as long as the contextual elements in it remain relevant to a given focus. Events that occur in any of the other spaces may lead some contextual elements to no longer influence the behavior of the STS structures as they perform their allocated functions. Therefore, as time goes forward and events happen, new situations are created. The time space is the sequence of situations that have been recorded for a given focus. A different focus will be associated to a different sequence of situations.

Keeping a record of the previous situations is important, as understanding context involves keeping "*a finite history of the time-space information on the state of the environment of the entities*" [25]. In other words, past situations of a focus are part of the context of the system, even though new situations have superseded them. This is because of the persistence property from Table 1: since contextual elements are interdependent, one situation may alter the state of one or more other contextual elements. When the situation is replaced by a new one, the effect it had on the altered contextual elements may still be active for a relatively long period of time (e.g. an emergency situation may have consequences that will be felt long after the end of the emergency itself).

A situational change may be triggered when one of the following events occurs (Figure 2):

1. An environment change has occurred, i.e., the state of the environment space is altered, thereby shifting the relevance of a subset of the contextual elements.
2. The value of a contextual element has changed beyond a certain threshold.
3. Either the structure or the function from the focus has changed, making the situation shift towards a different location within the environment space.

In other words, when a situational change occurs, this means that an event has occurred and has either altered the set of relevant contextual elements, or has modified the value of at least one contextual element that was already in the situation. In the latter case, the new value of the element must be sufficiently different from what it was before the event in order to have an influence on the operations of the STS of interest. The conditions for an event to happen can have their roots within the environment space itself, but also from modifications within the function or the structure space.

4 Example: air traffic control incident

We provide in this section a short example from the air traffic control domain that illustrates one application of the OSCAR framework. Consider the situational change from Figure 3.

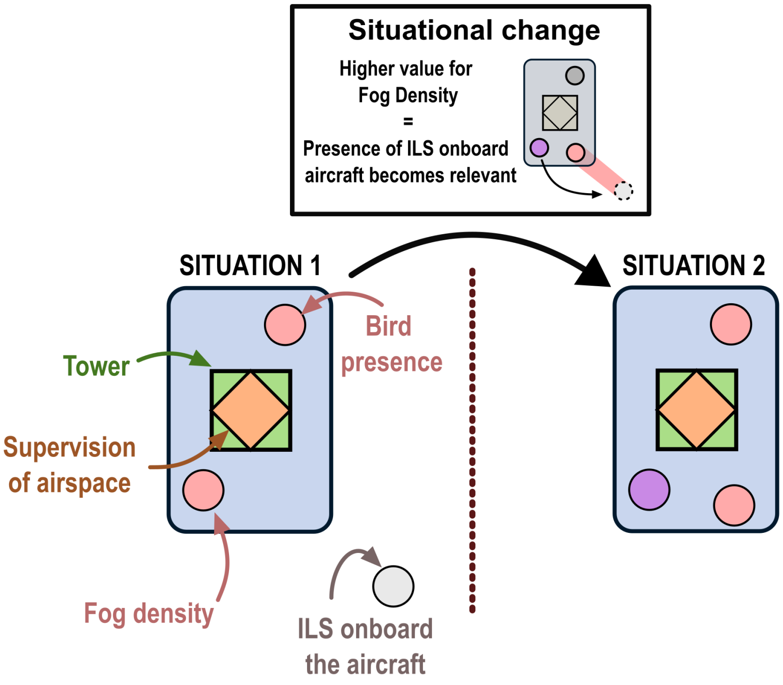


Figure 3: A situational change occurring with respect to the the tower/supervision focus

In this example, we chose as our focus the supervision of the airfield (the function) which is typically allocated to a physical control tower (the structure) located next to the runway. A typical situation for this focus (shown in the left of Figure 3) includes (but is not limited to) contextual elements such as the density of fog and whether there are obstacles like birds in the vicinity of the runway.

When the fog density is low enough, controllers and arriving aircrafts do not typically resort to Instrument Landing System (ILS) equipment that allow them to land with low visibility. Fog density is a positive real number defined on the volume around the tower and the runway, and it varies over time. When the density exceeds a minimum threshold, the visibility of controllers and pilots will become low enough to require them to use a specific category of ILS equipment (located both onboard the aircraft and on the ground). When ILS equipment is used, the situation becomes one in which Instrument Flight Rules (IFR) apply, making the landing procedures significantly different. Therefore, the capabilities of the aircraft regarding IFR will now matter: for instance, a terrain with very low visibility might lead a controller to redirect an aircraft not equipped with proper ILS equipment towards a different airfield.

In terms of the OSCAR framework, the increase in the fog density value triggers a situational change. This change leads the focus to get into a new situation (shown in the right of Figure 3). The "ILS onboard the aircraft" element has now entered this situation: this means that it has now become relevant to the focus, and thus its value (i.e., whether the aircraft is properly equipped with ILS or not) will drive future operations of the tower while achieving the airspace supervision function.

This example also illustrates that situational changes can be composed and rapidly succeed one another. Here, the functional architecture itself did not change simply because the fog density has increased slightly. It only changed when the fog density has increased so much that both controllers and pilots were no longer able to perform their operations as usual (this was the first situational change). Then, the functions from the function space were shifted and then reallocated to new structures (i.e., to ILS equipment) in order for the tower to continue to supervise the airfield, and for pilots to continue being cleared for landing (this was a second situational change).

5 Discussion

While the Air Traffic Control illustration from the previous section is primarily qualitative, its purpose is to demonstrate OSCAR's representational capabilities rather than to assert validated operational benefits. As a conceptual contribution, the framework necessarily abstracts from quantitative value metrics, and a formal demonstration of decision-making improvements would require substantial additional work. Given the complexity of the notions embedded within OSCAR, we expect that its real value can only be assessed in the long-term through multiple empirical studies. Meaningfully evaluating in an HSI-oriented way whether the framework reduces ambiguity or enhances human or system decision-making would require simulation or human-in-the-loop experimentation. These developments would constitute an important direction for subsequent work.

Furthermore, the OSCAR framework provides a novel conceptual foundation for representing operational context in STSs. Although its current formulation remains primarily descriptive, it helps clarify how context emerges from the dynamic interactions between structures, functions and their environments. By making these relationships explicit, OSCAR enables HSI practitioners to treat context as a *design variable* rather than a fixed external constraint that is only considered during the late stages of the system lifecycle.

5.1 Instantiating the OSCAR framework

To transition OSCAR from a conceptual model to an operational methodology, it must be instantiated through a set of systematic steps integrated within existing SE and modeling workflows. In this sense, the PRODEC methodology [8] provides a methodological framework that emphasizes the dynamic nature of operations, just as the OSCAR framework emphasizes the dynamic nature of context. Drawing inspiration from the PRODEC methodology [8], the instantiation process could be articulated as follows:

1. **Eliciting contextual elements:** Identify and formalize the contextual elements relevant to the system-of-interest through document analysis, field observations, and semi-structured interviews. This process results in a domain-specific ontology that populates the *environment space* of OSCAR.

2. **Defining structure–function focuses:** For each key system function, identify the structure(s) to which it is allocated and define the corresponding *focus* (structure; function). These focuses serve as reference points for the analysis of operational context, anchoring the *function* and *structure* spaces.

3. **Contextualizing scenarios:** Enrich operational scenarios with contextual elements and their expected variations to model *situations* and *situational changes*. These enriched scenarios instantiate the *time space* and make the evolution of context explicit.

4. **Simulating and validating:** Integrate these scenarios into simulation or human-in-the-loop environments to analyze how contextual changes propagate across the four spaces. This validation confirms whether the modeled context reflects operational reality and supports system-level decision-making.

5.2 OSCAR for Digital Twins of Socio-Technical Systems

The methodological approach described above echoes the rationale of the Reality Anchor Methodology (RAM) [26], where the concept of *reality anchors* (e.g. key perceptual elements that allow humans to perceive, comprehend, and project the operational situation) ensures that contextual representations are grounded in actual user needs and decision processes. Therefore, Digital Twins (DTs) of STSs represent a natural application domain for OSCAR, as they must integrate technical, human, and organizational data within a continuously evolving operational model. In this perspective, the reality anchors elicited through the RAM correspond directly to OSCAR's contextual elements, which are those environmental or systemic variables that influence how operators interpret and act upon the system state. Integrating OSCAR with RAM thus creates a unified design process where context modeling and decision support co-evolve.

In parallel to the necessity of operators to understand context to increase their situation awareness, the Framework for representing Intrinsic, Reflexive, Extrinsic dimensions of a DT (FIRE) [27] provides a reason to develop context models for DTs of STSs. FIRE conceptualizes the progression of DT from purely technical artifacts to socially embedded, self-aware systems capable of interpreting and adapting to their own context. In this sense, OSCAR provides the *contextual substrate* that reflexive systems require: a structured, dynamic representation of the world upon which adaptive reasoning can be built. Combining OSCAR's explicit context representation with FIRE's reflexive modeling principles could enable DTs not only to mirror operational reality but also to understand and reason about it—an essential step toward resilience and autonomy in complex, human-centered systems.

6 Conclusion

In this paper, we have proposed the OSCAR framework in order to better understand the relationship between a system and its context. We have been interested in particular in the class of sociotechnical systems, as those are deeply rooted in an environment that affects their behavior during operations. Indeed, the operational complexity of STSs arises from the intricate relationships between both the internal and external structures that are at some point involved in the system lifecycle. The framework is aimed at providing SE and HSI practitioners a tool to explore the critical role of context in the design and analysis of such systems.

The OSCAR framework provides a dynamic, multi-scale view of context, integrating the function, structure, environment and time spaces. The framework offers a mechanism for understanding how changes in one space affect others. In this regard, we have identified three factors of situational change, depending on the nature of events occurring within the function, structure or environment spaces.

Future research work includes framework validation through practical application to real-world STS case studies. Validation should also lead to further refinement of the nature of the interactions between the spaces, and between contextual elements themselves. We also plan to harness the framework and combine it with the HSI literature about STS engineering in order to develop an early-stage design software modeling environment that would enable designers to integrate operational design scenarios with dynamic contextual information description. This environment would support engineering principles from HSI and enrich the set of model-based tools available to systems engineers and complex system designers.

References

[1] J. Ruault, F. Vanderhaegen, and D. Luzeaux, "Sociotechnical systems resilience," in *INCOSE International Symposium*, vol. 22, 07 2012.

[2] NASA, *Human Systems Integration Handbook.* NASA, 2021.

[3] G. A. Boy, *Human–Systems Integration: From Virtual to Tangible.* Boca Raton: CRC Press, Jan. 2020.

[4] G. Baxter and I. Sommerville, "Socio-technical systems: From design methods to systems engineering," *Interacting with Computers*, vol. 23, no. 1, pp. 4–17, 08 2010. DOI:10.1016/j.intcom.2010.07.003

[5] E. Crawley, B. Cameron, and D. Selva, *System Architecture: Strategy and Product Development for Complex Systems*, 1st ed. USA: Prentice Hall Press, 2015.

[6] B. Edmonds, "Complexity and context-dependency," *Foundations of Science*, vol. 18, no. 4, pp. 745–755, Nov 2013. DOI:10.1007/s10699-012-9303-x

[7] A. Disdier, D. Masson, M. Jankovic, and G.-A. Boy, "Defining and characterizing the operational context for human systems integration," *Systems Engineering*, vol. 27, no. 6, pp. 1103–1115, 2024. DOI:10.1002/sys.21775

[8] G. A. Boy, D. Masson, E. Durnerin, and C. Morel, "PRODEC for human systems integration of increasingly autonomous systems," *Systems Engineering*, vol. 27, no. 5, pp. 805–826, 2024. DOI:10.1002/sys.21751

[9] J. L. Austin, *How to do things with words*, ser. William James Lectures. Oxford University Press, 1962.

[10] J. McCarthy, "Notes on formalizing context," in *Proceedings of the 13th International Joint Conference on Artifical Intelligence - Volume 1*, ser. IJCAI'93. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993, p. 555–560.

[11] M. Daun and B. Tenbergen, “Context modeling for cyber-physical systems,” *Journal of Software: Evolution and Process*, vol. 35, no. 7, p. e2451, 2023. DOI:10.1002/smr.2451

[12] INCOSE, *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, 5th ed. John Wiley and Sons, 2023.

[13] SEBoK Editorial Board, *The Guide to the Systems Engineering Body of Knowledge (SEBoK)*, 2nd ed. Hoboken, NJ: The Trustees of the Stevens Institute of Technology, 2023.

[14] H. Allert and C. Richter, “Practices, Systems, and Context Working as Core Concepts in Modeling Socio-Technical Systems: Fifth International Workshop on Philosophy and Informatics,” *Proceedings of the Fifth International Workshop on Philosophy and Informatics (WSPI 2008)*, vol. 332, pp. 19–33, 2008.

[15] P. Brézillon, “Context modeling: Task model and practice model,” in *Modeling and Using Context*. Springer, 2007, pp. 122–135.

[16] D. Hitchins, *Systems Engineering: A 21st Century Systems Methodology.* John Wiley & Sons, 2007.

[17] R. L. Ackoff, “Towards a system of systems concepts,” *Management science*, vol. 17, no. 11, pp. 661–671, 1971.

[18] K. M. Feigh and A. R. Pritchett, “Requirements for Effective Function Allocation: A Critical Review,” *Journal of Cognitive Engineering and Decision Making*, vol. 8, no. 1, pp. 23–32, Mar. 2014. DOI:10.1177/1555343413490945

[19] G. Pahl, W. Beitz, J. Feldhusen, and K.-H. Grote, *Engineering Design: A Systematic Approach.* Springer, 01 2007.

[20] M. Hause and M. Wilson, “Integrated human factors views in the unified architecture framework,” *INCOSE International Symposium*, vol. 27, no. 1, pp. 1054–1069, 2017. DOI:10.1002/j.2334-5837.2017.00412.x

[21] D. W. Orellana and A. M. Madni, “An architecture profile for human–system integration,” in *Disciplinary Convergence in Systems Engineering Research*, A. M. Madni, B. Boehm, R. G. Ghanem, D. Erwin, and M. J. Wheaton, Eds. Cham: Springer International Publishing, 2018, pp. 395–406.

[22] C. Gerber, J. Siekmann, and G. Vierke, “Holonic multi-agent systems,” DFKI, workingPaper, 1999.

[23] L. S. Wheatcraft, “Everything you wanted to know about interfaces, but were afraid to ask,” in *INCOSE international symposium*, vol. 20, no. 1. Wiley Online Library, 2010, pp. 1132–1149.

[24] B. Stensrud, G. Barrett, and A. Gonzalez, “Context-based reasoning: A revised specification,” in *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2004)*, 2004, pp. 603–610.

[25] N. Nwiabu, I. Allison, P. Holt, P. Lowit, and B. Oyeneyin, “Situation awareness in context-aware case-based decision support,” in *2011 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, 2011, pp. 9–16.

[26] S. Camara Dit Pinto, E. Villeneuve, D. Masson, G. A. Boy, and L. Urfels, “Reality anchor methodology: How to design a digital twin to support situation awareness,” *Journal of Engineering Design*, vol. 35, no. 6, p. 775–799, 2022. DOI:10.1080/09544828.2024.2336838

[27] D. Masson and E. Villeneuve, “Fire: A human-centered framework for digital twin design,” *Systems Engineering (in press)*, 2025. DOI:10.1002/sys.70014