A Hierarchical Framework for Safety in Orchestrated Systems-of-Systems

Julieth Patricia Castellanos Ardila & Sasikumar Punnekkat & Gabriele Capannini & Sara Abbaspour
Mälardalen University- Västerås, Sweden
{julieth.castellanos & sasikumar.punnekkat & gabriele.capannini & sara.abbaspour}@mdu.se

I. Introduction

A System of Systems (SoS) [1] comprises independent constituent systems (CS) pursuing a shared goal, such as mixed traffic in underground mining [2] or mass removal in construction [3]. An SoS may exhibit emergent behaviors from CS interactions and dependencies that cannot be inferred from individual CS [4]. Such behavior complicates SoS-level analysis and management, motivating coordination mechanisms such as orchestration [5]. In safety-critical SoS, orchestration introduces additional challenges because compromised data or control flows may invalidate operational assumptions and lead to unsafe decisions. Therefore, new approaches are needed to analyze SoS-level properties hile accounting for the combined effects of CS interactions, orchestration, AI-enabled decisions, interconnectedness, and human oversight.

In this position paper, we propose CHORUS, a **C**onceptual **H**ierarchy for **O**rganizing, **R**easoning, and **U**nderstanding **S**oS properties. CHORUS takes inspiration from Bounding Volume Hierarchies (BVH) [6], which organize spatial entities into nested regions and allow non-interacting regions to be considered separately. Used as an organizing principle, CHORUS supports separation of concerns by partitioning the SoS into bounded areas of responsibility and focusing detailed reasoning at the level where relevant interactions may occur.

This principle is mapped to hierarchical SoS levels described in [7]. The micro-level concerns individual CS. The meso-level concerns constellations, i.e., subsets of CS grouped by a shared task, operational area, or interaction context, that can operate largely independently of other constellations. The macro-level coordinates and synchronizes these constellations and handles cross-partition dependencies, including the reassignment of CS and the communication of events whose effects extend beyond their originating constellation.

In this instantiation, CHORUS treats safety as a systemic property of an orchestrated SoS. Safety constraints and functions are allocated according to whether they concern individual CS, interactions within a constellation, or coordination across constellations. Cross-cutting reinforcement mechanisms grounded in defense-in-depth [8] support these functions by protecting assumptions, detecting deviations, constraining AI-enabled decisions, and enabling intervention. The safety instantiation is illustrated through a construction scenario.

This work was supported by MDU's TSS initiative through project DOSTY (Dependable Collaboration of Intelligent System of Trusted Smart Systems).

II. A Hierarchical Safety Perspective

CHORUS begins by identifying SoS-level hazards and deriving safety constraints, for example, through Systems-Theoretic Process Analysis (STPA) [9]. It then examines the interactions through which these constraints may be violated and allocates the corresponding safety functions to the appropriate hierarchical level. Following the BVH-inspired principle, detailed reasoning is restricted to configurations in which relevant interactions or unsafe conditions may arise. The method comprises two complementary elements: hierarchical safety functions and cross-cutting reinforcement mechanisms.

A. Hierarchical Safety Functions

Safety functions are allocated across the CHORUS hierarchy based on the scope of the unsafe condition identified through the preceding hazard analysis and the responsibilities required to address it. The hierarchy supports the allocation of each function to the level at which it can be most effectively implemented and enforced: an individual CS, interactions within a constellation, or coordination across constellations.

1) **Micro-level:** Safety functions target individual CS. They address causal factors that can be handled locally through the CS's own sensing, control and actuation.
2) **Meso-level:** Safety functions target interactions within a constellation. They address causal factors that arise when CS, although individually safe, interact or coordinate in conflicting ways within a shared task, operational area, or interaction context. These functions can be enforced locally without unnecessarily affecting other constellations.
3) **Macro-level:** Safety functions target the orchestration logic. They address causal factors introduced or amplified by planning, scheduling, task allocation, prioritization, or adaptation of the SoS as a whole.

B. Cross-Cutting Reinforcement Mechanisms

Cross-cutting reinforcement mechanisms support safety functions across CHORUS by protecting assumptions that may be invalidated during operation. These assumptions concern trustworthy data and control flows, expected behavior, acceptable AI-enabled decisions, and available intervention when automation is uncertain or degraded. Cyberattacks, such as spoofing or denial of service, may undermine these assumptions and compromise safety functions. Reinforcement mechanisms therefore do not replace safety functions but help maintain their effectiveness. Examples include:

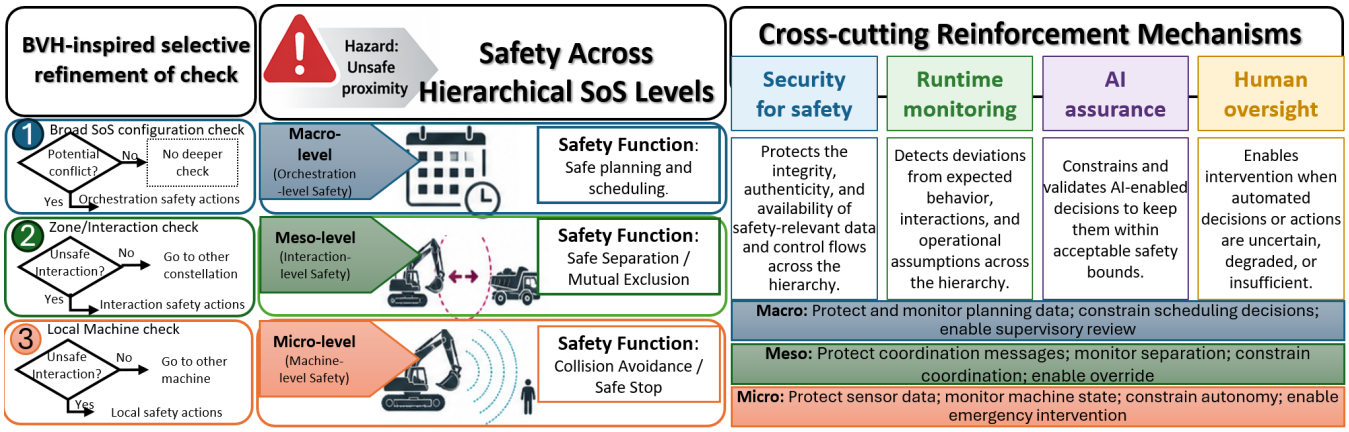

Fig. 1. CHORUS Safety Instantiation for Unsafe Proximity in an Orchestrated Construction SoS

a) **Security for safety:** Protects the integrity, authenticity, and availability of data and control flows against cyber-attacks that could cause unsafe orchestration decisions.
b) **Runtime monitoring:** Checks whether operational assumptions remain valid and detects deviations in CS behavior, interactions, or orchestration decisions.
c) **AI assurance:** Constrains and validates AI-enabled decisions that may affect safety, including perception, prediction, planning, coordination, and orchestration.
d) **Human oversight:** Enables intervention when automated mechanisms are uncertain, degraded, or insufficient

III. Unsafe Proximity in Construction SoS

Figure 1 illustrate how CHORUS can structure the analysis of unsafe proximity in an orchestrated construction SoS [3]. Starting from the SoS-level hazard, potential conflicts are progressively examined at the constellation and machine levels. Safety functions are then assigned where the corresponding unsafe condition can be controlled, e.g., collision avoidance and safe stop at the micro-level, safe separation and mutual exclusion at the meso-level, and safe planning and scheduling at the macro-level. The levels are connected bidirectionally. Macro-level plans constrain constellation coordination and machine operation, while detected conflicts, degraded capabilities, or failed assumptions propagate upward and may trigger rescheduling, reassignment, or supervisory intervention.

Applied in practice, the analysis therefore follows four steps: identify an SoS-level hazard and its constraints; locate the interactions through which it may arise; allocate safety functions to the responsible hierarchical levels; and select reinforcement mechanisms for assumptions that require protection, monitoring, assurance, or intervention. In the construction case, manipulated position data may first be detected at the micro-level, invalidate separation decisions at the meso-level, and ultimately require macro-level rescheduling. BVH-inspired refinement limits detailed analysis to configurations and interactions in which conflicts are possible, rather than treating every CS combination as equally relevant.

IV. Conclusion and Outlook

As a position paper, this work introduces CHORUS as a conceptual framework rather than a validated analysis method. Its usefulness therefore remains to be established empirically. Planned evaluation will compare CHORUS-supported and conventional SoS safety analyses in terms of identified interaction hazards, traceability of safety responsibilities across levels, analysis effort, and support for assurance arguments. Current limitations include dependence on the quality of the initial hazard analysis, possible ambiguity when a safety function spans multiple levels, and the cost of maintaining the hierarchy as constellations and responsibilities change. Scalability is expected from BVH-inspired pruning of irrelevant interactions, but highly coupled SoS may provide fewer opportunities for such separation. Future work will formalize cross-level relationships, define allocation and escalation criteria, and evaluate CHORUS through industrial case studies and comparison with existing SoS engineering approaches.

References

[1] M. W. Maier, "Architecting Principles for Systems-of-Systems," *Systems Engineering*, vol. 1, no. 4, pp. 267–284, 1998.
[2] J. P. Castellanos-Ardila, S. Punnekkat, H. Hansson, and P. Backeman, "Safety argumentation for machinery assembly control software," in *43rd SafeComp Conference*. Springer, 2024, pp. 251–266.
[3] N. Ali, M. Naeem, J. P. Castellanos-Ardila, and S. Punnekkat, "Safety-Aware Strategy Synthesis for Autonomous System of Systems with UPPAAL," in *44th SafeComp Conference*. Springer, 2025, pp. 73–87.
[4] C. W. Johnson, "What are Emergent Properties and How do They Affect the Engineering of Complex Systems?" *Reliability Engineering and System Safety*, vol. 91, no. 12, pp. 1475–1481, 2006.
[5] T. Nordstrom, L. R. Sutfeld, and T. Besker, "Exploring different actor roles in orchestrations of system of systems," in *19th SoSE Conference*, 2024, pp. 190–196.
[6] C. Ericson, *Real-Time Collision Detection*. San Francisco, CA: Morgan Kaufmann, 2005.
[7] J. Axelsson, "A Refined Terminology on System-of-Systems Substructure and Constituent System States," in *14th SoSE Conference*. IEEE, 2019, pp. 31–36.
[8] International Nuclear Safety Advisory Group, "Defence in depth in nuclear safety," International Atomic Energy Agency, Vienna, Austria, Tech. Rep. INSAG-10, 1996.
[9] N. G. Leveson and J. P. Thomas, *STPA Handbook*. Cambridge, MA: Massachusetts Institute of Technology, 2018.