

How Systems Change: A Preliminary Ontological Account

Rodrigo F. Calhau¹, João Paulo A. Almeida²,
Giancarlo Guizzardi³, and Victorio A. Carvalho¹

¹ LEDS Research Group, Federal Institute of Espírito Santo, Brazil
{calhau, victorio}@ifes.edu.br

² Ontology & Conceptual Modeling Research Group (NEMO),
Federal University of Espírito Santo, Brazil
jpalmeida@ieee.org

³ Semantics, Cybersecurity & Services, University of Twente, The Netherlands
g.guizzardi@utwente.nl

Abstract. Systems are usually conceived of as ‘complex’ (or ‘organized’ wholes) composed of ‘connected’ (or interacting) elements. They are dynamic entities whose composition, structure, aspects, and environment may change over time. In this paper, we take a step towards ontological foundations for modeling system change. Building on the Unified Foundational Ontology (UFO), we account for different dimensions of system variation over time inspired on Bunge’s CESM model. The proposed foundations contribute to system change understanding and provide a conceptual base for future applications in system modeling approaches and languages.

Keywords: System · System change · System evolution · System situation · Ontology · Systems modeling

1 Introduction

Systems such as organizations, software systems, and cyber-physical systems are inherently dynamic entities that undergo continuous structural, compositional, environmental, and aspectual changes over time [1, 4, 8]. Such changes may involve the addition or removal of components, changes in their relationships, the emergence or dissolution of aspects, the variation of qualities and capabilities, and as a result of the latter, modifications in the possible interactions between the system and its environment. To properly understand and model these different dimensions of change in fields such as systems engineering, software engineering, and enterprise architecture, we need a proper understanding of this phenomenon.

This work proposes a preliminary account of system change by extending a core system ontology proposed in our previous work [5–7]. Our aim is to provide an ontological foundation for system change based on the Unified Foundational Ontology (UFO) [9] and inspired by Bunge’s CESM model [4].

The remainder of this paper is organized as follows. Section 2 presents the ontological baseline covering briefly the foundational ontology and the *system core ontology* [5–7],

both of which are employed in Section 3 to develop the ontological foundation for system change; Section 4 discusses implications in system modeling, and presents final considerations and future works.

2 Ontological Baseline

Unified Foundational Ontology (UFO). UFO [9] is a Foundational Ontology, i.e., an axiomatic system of domain-independent ontological theories. UFO classifies entities into *individuals* and *types*. Individuals can be *situations*, *events*, and *endurants*. *Situations* are specific arrangements of a particular portion of reality. *Events* are individuals that “occur” or “happen” across time (e.g., the FIFA 2026 World Cup Final). *Endurants* are individuals which persist in time and may undergo qualitative changes without losing their identity. *Endurants* fall into two categories: *objects* and *aspects*. *Objects* are endurants that exist independently (such as the Moon, the Statue of Liberty). Of particular importance to account for systems is the notion of *functional complex*. A *functional complex* is an object composed of multiple parts that play functional roles with respect to the whole (e.g., an organization, the human body, a computer). Aspects, on the other hand, depend on other individuals to exist. A *relator* is a specific type of *aspect* that can mediate two or more things (e.g., John Lennon and Yoko Ono’s marriage). Material relations are those derived from relator types (e.g., in the way “being married to” is derived from Marriage, or “enrolled at” is founded on Enrollment). In contrast to a relator, an *intrinsic aspect* is existentially dependent on a single individual. An intrinsic aspect can be a *mode* or a *quality*. *Qualities*, like color, height, weight, etc., may be directly projected into a value space. *Modes* are dispositional entities that are manifested through events, and which can have aspects of their own.

In contrast to *individuals*, *types* are entities that define common properties that are shared by their instances. The UFO taxonomy of types is organized into *endurant types* (e.g., “Car”, “Partnership”), *event types* (e.g., “Live Concert”, “Birth”), *object types* (e.g., “Person”, “Organization”), *aspect types* (e.g., “Age”, “Enrollment”), *situation types* (e.g., “Battery Level is Low”), etc. Endurant types are further classified according to their modal metaproperties (concerning rigidity and sortality). Rigid types, like *kinds* (e.g., “Car”) and *subkinds* (e.g., “Ferrari”), are those that apply to their instances necessarily (and hence, apply throughout their instances existences). Anti-rigid types are those that classify their instances “dynamically”. These include *roles* and *phases*. *Roles* are relationally-dependent (“Student”, “Spouse”) and *phases* classify their instances according to (mutable) intrinsic characteristics, e.g., “WorkingCar”, “BrokenCar”. Sortals are those endurant types that supply or carry a principle of identity for their instances (e.g., the object kind “Car”, “Person”, and the relator kind “Marriage”), while non-sortals are those (also called *dispersive types*) which abstract common features of sortals which different identity criteria (e.g., the category “Physical Object” is a rigid type subsuming “Car” and “Person”). Non-sortals that apply necessarily to some instances but contingently to others are called *mixins*.

Core Ontology of Systems (COSy). COSy⁴ is a core ontology inspired by the CESM model [4] and grounded on the Unified Foundational Ontology (UFO). A portion of this

⁴ <https://github.com/nemo-ufes/cosy>

3 Ontological Foundations for System Change

In this ontology, systems are considered entities subject to change, i.e., that can vary over time. When a system changes, it changes to a new *situation* – a certain configuration of parts and properties in a given time interval. This idea is illustrated in Figure 2: the system s is formed by four system situations: ss_1 (from t_0 to t_1), ss_2 (from t_1 to t_2), ss_3 (from t_2 to t_3), and ss_4 (from t_3 to t_4), each corresponding to a different configuration of s in a different period of time.

Figure 3 reveals *system situation* as a complex situation formed by the following parts: *Compositional Situation*, *Structural Situation*, *Environmental Situation* and *Aspectual*

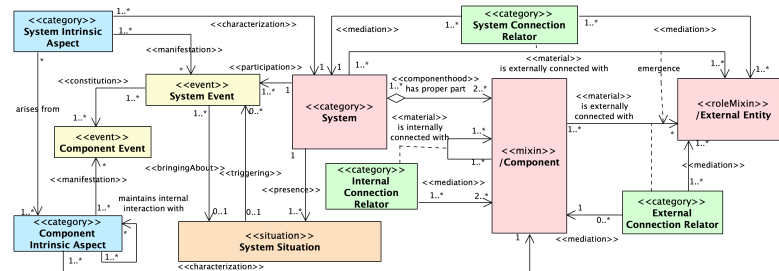


Fig. 1. Fragment of *COSy*: a Core Ontology of Systems, proposed in [5–7]

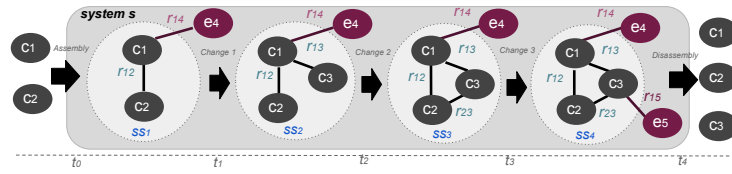


Fig. 2. An illustration of a system changing across different situations (configurations)

Situation, which are inspired in Bunge’s *CESM* (*Composition, Environment, Structure, Mechanism*) model. In the original model, the *composition*(t) of system s is formed by the set of components that are parts of s at time t ; the *environment*(t) is formed by all external entities that are connected to system s at time t ; the *structure*(t) of system s is formed by all the connections between the components (internal or external) of system s at time t ; finally, the *mechanism* is formed by the types of “processes” or “events” in which system s can participate. In what follows, these different dimensions of the CESM model are reconstructed and explained in terms of UFO elements.

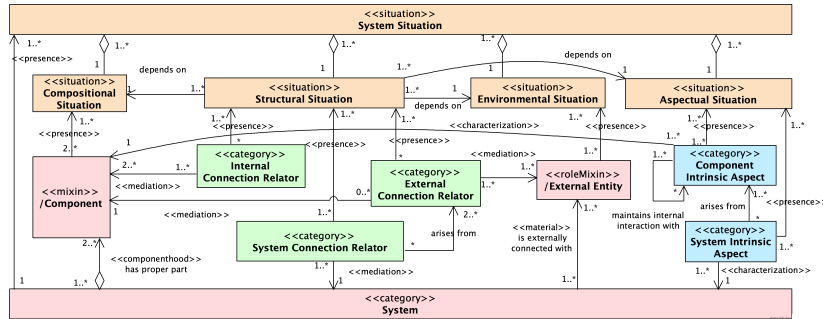


Fig. 3. An extension to the COSy Ontology with types of situations representing the CESM dimensions in which a system can change

Compositional Change. At a particular time interval, the components of a system stand in what we term *Compositional Situation*. For example, suppose system s shown in Figure 2 is a “supervision system” initially composed of two components, the supervisor c_1 and the supervisee c_2 , its composition in the period between t_0 and t_1 , in system situation ss_1 . With changes in the system, new situations are generated and new compositions can arise when components are added to or removed from the system. System s undergoes a change at time t_1 (Fig. 2) in which a new component c_3 (a new supervisee) is added, generating a new composition consisting of c_1 , c_2 , and c_3 , and hence, a new compositional situation.

Structural Change. A system structural situation includes the (system and component) *connection relators* which reify the (internal and external) connections existing in a

system at a given time interval. Consider the supervision system s in situation ss_2 (Fig. 2). At this moment, the system is formed by three components: a supervisor (c_1) and two supervisees (c_2 and c_3). The supervisor has a “supervision” connection with each of the two supervisees (who initially are not related). These (internal) connections are mediated by the relators r_{12} and r_{13} , as illustrated. Supervisor c_1 is also (externally) connected to an external researcher e_4 , through relator r_{14} . Thus, the structure of the system in the situation ss_2 is formed by r_{12} and r_{13} , corresponding to the supervision connections between the supervisor and the supervisees, and by r_{14} corresponding to an external connection. Just as the composition of a system can “change”, so can the structure. In the example, supervisees c_2 and c_3 are connected only to the supervisor c_1 in situation ss_2 . However, suppose that the two supervisees c_2 and c_3 start collaborating with each other as well. As a result, a new collaboration connection arises between them, mediated by the relator r_{23} (Fig. 2). Due to this change, a new system structure is generated in system situation ss_3 , formed by the (internal) relators r_{12} , r_{13} , and r_{23} , and by the (external) relator r_{14} .

Aspectual Situation. The aspectual situation corresponds to a situation formed by (i) system and component aspects, including *intrinsic modes* and *qualities*; (ii) corresponding *quality values* that a system’s (and components’) aspects possesses in a given period of time. This is shown in Figure 3, in which *component aspects* and *system aspects*, from a same system, stands in a *aspectual situation*. Examples include the *fuel level* of a car at a given moment, the *geographical position* of this car, and the *engine temperature* at that same moment. In the case of system s (Fig. 2), in situation ss_2 , there is no collaboration connection between c_2 and c_3 because the level of collaboration capability, cc_2 and cc_3 , of each supervisee is low. Consider the system undergoes an aspectual change and the collaboration capability cc_2 and cc_3 of each supervisee c_2 and c_3 increase, generating a new aspectual situation in situation ss_3 .

Environmental Situation. In a system environmental situation, we have standing in all external entities connected to the system Figure 3. In the example of system s (Fig. 2), the external researcher e_4 is an example of an external entity that is part of the environment in the situation ss_3 . Similarly to the other types of situations, a “change” can transform an environmental situations to another. In the example of system s , the environment of system s in situation ss_3 is formed by the external entity e_4 . As the system undergoes a change, situation ss_4 is brought about, including the external collaborator e_5 connected to the system through the supervised c_3 .

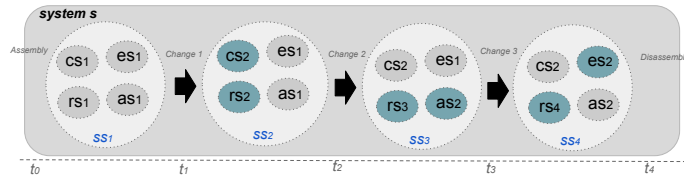


Fig. 4. System Change - Illustration (cs_x : compositional situation; rs_x : structural situation; es_x : environmental situation; as_x : aspectual situation)

Multidimensional Change. An important aspect is that although the previous examples focused only on changes in a single dimension, changes generally occur in more than one dimension of the system. This is because all these dimensions of a system are directly or indirectly interconnected. For example, the (internal) structure of a system is defined on the basis of its composition. Thus, a change in the composition of a system can cause a change in its structure. Analogously, changes in the external structure of a system and in its environment can mutually influence each other. The structure is also based on the aspectual situation, since the system’s connections result from the (dispositional) aspects of the components. In short, the structural situation “depends” on compositional, environmental, and aspectual situations (Fig. 3). Thus, changes in these dimensions of the system will impact its structure. Figure 4 illustrates the dynamics of multidimensional change in the system’s situations, from the running example. As illustrated, change 1 occurs because there is a change in composition (cs_1 to cs_2) which triggers a change in the system’s structure (from rs_1 to rs_2), generating the situation ss_2 . Change 2 occurs due to a change in aspectual situation (as_1 to as_2) which also triggers a change in structure (rs_2 to rs_3), generating the system’s ss_3 situation. Change 3 occurs due to a change in environment (es_1 to es_2) which again triggers a change in structure (rs_3 to rs_4), generating the system’s ss_4 situation.

System Change Event. Based on the previous definitions of system situations, we consider a system change to be an event (in UFO terms) that generates a new system situation. System events are triggered by system situations, and generate another system situation as an outcome, except for assembling and disassembling events (Fig. 5). Based on the dimensions presented earlier, we define two main types of change in a system: *aspectual change* and *configuration change*. The first type corresponds to a change in the “state” of the system, that is, a change in the values of the aspects (properties) of the system. For example, a car can be turned on, turned off, accelerated, braked, etc. The second type of change, configuration change, involves change in at least one non-aspectual dimension, namely, composition, structure, and environment. Consequently, a configuration change may also involve a change in the system’s aspectual situation, but an aspectual change does not necessarily involve direct changes to composition, structure, and environment. Examples include changing a tire on a car, adding a tracking device to the car, or replacing a faulty radiator. Systems are composed of components, and therefore, system events are constituted by component events as well (Fig. 5). This

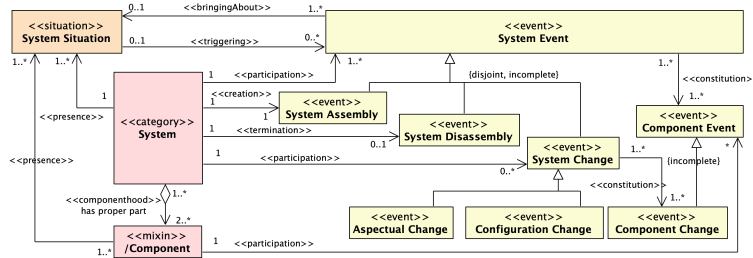


Fig. 5. System Situation Sub-Ontology Fragment: System Change (in OntoUML)

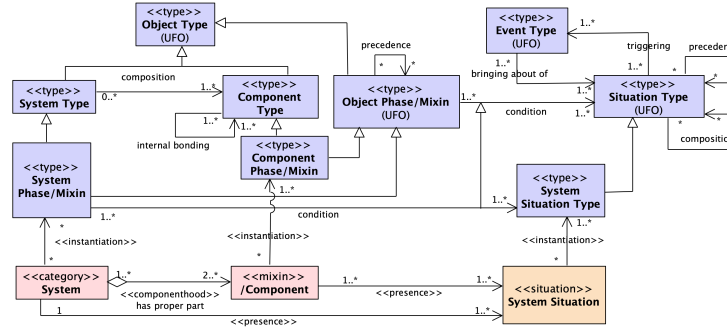


Fig. 6. System Situation Type - Sub Ontology (in OntoUML)

means that a system change necessarily also involves changes in its parts, including the relationships between them, and also in the aspects of the system or components.

System Situation Type Sub-Ontology. We also present a sub-ontology at the type level as illustrated in Figure 6. This sub-ontology is based on some essential UFO distinctions such as *object type*, *event type*, *situation type*, and *object phase / mixin* (an *object type* that specialize a *phase* or *phaseMixin*). As shown, instances of an *event type* ‘bring about of’ instances of a *situation type*, which instances, in turn, ‘trigger’ an *event type* instances. *Object types* can have different (sub) types of *object phase/mixin*, that include both *object phase* (with just one identity criteria) and *object phase mixin* (with distinct identity criteria). *Object phases/mixin* are, then, anti-rigid types that classify instances of *object types* dynamically, based on intrinsic and relational conditions represented by situation types. In this way, *object phases/mixin* have *situation types* as a condition, as illustrated. For example, a car instantiates the phase “fueled” if its situation instantiates the situation type “the car has gas level greater than zero”. In summary, *object phases/mixin* are dynamic classifications of *objects* (e.g., systems) based on *situation types*, which are universal situations that condition the instantiation of such *object phases/mixin*. *Object types* include *system types* and *component types*, which are linked by a ‘composition’ relation [7]. So, as a result, *object phases/mixin* include *system phase/mixins* as well as *component phases/mixins*. The *system phase/mixins* have the *component phase/mixins* as ‘condition’, as illustrated. For example, the “out of gas” phase of car type depends on the “empty” phase of the gas tank type. A *system phase/mixin*, as a phase, also has *system situation types* as conditions. For example, the “out of gas” phase of car type has the “tank with no gas” system situation type as a condition. This means, a car can just instantiate the “out of gas” phase if its gas level is zero, i.e., it instantiates different phases based on the current system situation.

Ontology Illustration. We apply here the ontology to model the *robot vacuum cleaner* scenario. This system comprises a number of components, such as a rotating brush (br_1), dustbin (db_1), battery (ba_1), main circuit board (cb_1), sensor system (se_1), among others not shown, such as wheels, propulsion motor, etc. The robot can have different phases: at rest, cleaning, at standby, stuck or with an error. The components also have phases. For example, the battery can be *full* or *dead* (static phase); the battery can

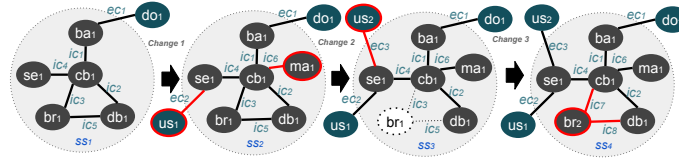


Fig. 7. Robot Vacuum Cleaner Situations Illustration

also be *charging* (when it is in the dock) or *discharging*, when it is in use (active phase); the brush can be *good for use* or *worn*; the dustbin can be *empty* or *full*. The instantiation of these phases depends on aspectual conditions and quality values associated with the robot and its components. The robot has qualities like its current position, a map of the environment, etc. The robot's components have aspects as: the dustbin has a dirt level; the battery has a charging level; the brush has a usage time, meaning it can be good for use or worn out (needing to be replaced). Regarding the robot's basic structure, the components mentioned above are connected to the main circuit board (cb_1). Concerning the environment, the robot interacts with different external entities such as the user (us_1)—directly or through a smartphone—, the dock (do_1), and obstacles. Figure 7 shows different changes undergone by the robot vacuum cleaner (ro), focusing on *configuration changes*. Changes in aspects will not be illustrated in the figure but will be addressed in the text. In the robot's structure in system situation ss_1 (Fig. 7), the rotating brush (br_1), dustbin (db_1), battery (ba_1), and sensor system (se_1) components are connected to the main circuit board (cb_1). In this situation (configuration), the robot has one external entity as part of its environment, the dock (do_1), connected to the robot's battery (ba_1). As depicted, a configuration change from situations ss_1 and ss_2 occurs in the system since the composition, environment, and structure of the system are modified. In this case, a mobile application is configured to access and control the robot vacuum cleaner remotely, and as a result, the user (us_1) becomes connected to the robot through its mobile application (environmental and structural change). In addition, the mop module (ma_1) is added to mop the house (compositional and structural change).

Concerning the aspectual situation, robot operation generates multiple aspectual changes, and creating distinct aspectual situations and corresponding phase transitions. For example, the robot's battery is charged (aspectual change) by placing it in the dock, and battery get full (phase transition); robot changes its phase from resting to cleaning (phase transition) and starts cleaning the environment with the activation event, at a predefined time; robot returns to the dock with lower battery level and the higher dirt level after finishing cleaning (aspectual change) and start sleeping (phase transition), and so on. Figure 8 illustrates an example of an aspectual change in the robot. After cleaning the house, the battery level drops to zero (system situation) and the robot's battery becomes empty (component phase). Consequently, the robot becomes discharged (system phase), triggering a recharge (system event) that brings the battery level to full (system situation). As a result of this aspectual change, the robot becomes charged.

These aspectual changes affect only aspect values (e.g., robot position, battery level, level of dirt in the compartment, etc.), without modifying necessarily system configu-

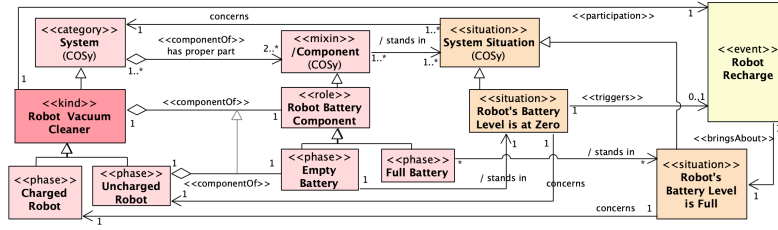


Fig. 8. Extending *COSy* to Illustrate the Robot Vacuum Cleaner Phases, Situations and Aspectual Change (in OntoUML)

ration (composition, structure, or environment). However, aspectual changes can also stimulate changes in the system configuration. Suppose the brush exceeds 300 hours of use, surpassing its limit. The brush usage time changes (aspectual change), then it becomes worn (phase transition) and is blocked by the system, ceasing to respond to commands and interact with other components. This is shown in Figure 7 where, in the situation ss_3 , the brush (br_1) loses connection with the other components and becomes non-functional (structural change). As illustrated, the new situation ss_3 also included a new user (us_2) that becomes an external entity to the robot (environmental change). The brush degradation then induces another system configuration change involving the brush replacement (compositional and structural change). This change is made, in which the brush (br_1) is replaced by a new brush (br_2), generating a new situation ss_4 . Brush replacement restores the robot's operational capability in situation ss_4 . This example illustrates bidirectional dependency between aspectual and configurational changes.

4 Implications and Final Remarks

The proposed ontology provides ontological foundations for integrating these perspectives across multidimensional systems situations and change events. Class diagrams in UML or block/part diagrams in SysML generally focus on the structural perspective in isolation. Similarly, state-based diagrams in UML or SysML generally lack proper integration with other perspectives. Thus, when a change occurs in the system represented in a state diagram, it is unclear what the impacts of such a change are on the structure, composition, etc. In many state-based modeling approaches, states are represented as atomic labels, without an explicit ontological commitment to the underlying situation (e.g., configuration) of the system. The ontology enables to relate a phase of a system to a specific structure or configuration of the system. With the proposed ontology, it is possible to ground the phases of the system based on different types of situations proposed in this work. For example, a system cannot instantiate a “cleaning” phase if the cleaning brush is disconnected. Another important aspect is related to the term “state”. In notations like UML/SysML, the meaning of “state” is ambiguous. As presented, systems instantiate phases that are grounded on aspectual or configurational situations of the system. That is, what is called “state” in such notations can actually be mapped to the system's phase, which is a dynamic classification of it grounded on its current

situation. Furthermore, the ontology also allows to understand how phases of the part and the whole relate, enabling improved precision and expressiveness in state-based models. Beyond this integration of simultaneous phases, the type level, with its proposed hierarchy of phases, contributes to a global understanding of system behavior, allowing for the creation of taxonomies or hierarchies of system phases. In general, the proposed ontology contributes to a more integrated and semantically grounded representation of system change, allowing for explicit relationships between structure, behavior, environment, and aspects in distinct system situations.

Final Remarks. This work presented an ontological foundation for system change. System situations were categorized based on different dimensions of a system, drawing on Bunge’s work. This categorization served as a basis for understanding different types of system change involving one dimension, such as aspectual changes, or multidimensional changes, such as configurational changes. The ontology also addressed the type level, where phases of a system were approached based on situation types. One aspect that needs further development concerns the system’s and component’s function, which can change. We implicitly included the system’s function within the system situation, but in the future it may give rise to an exclusive sub-situation for the *functional situation*. Our focus in this work was greater on static situations (e.g., configuration); however, a relevant future work is to better explore ongoing “behavioral situations in systems” (while they manifest a capability), relating to active phases of the system, e.g., the performance of tasks and activities of a system. Finally, in this work we briefly addressed relationships between situations; however, it would be interesting to conduct a more in-depth investigation in the future into how situations relate ontologically. In general, the proposed ontology provides a basis for understanding system change as multidimensional and time-based, enabling future investigations into the ontological dependencies between emergent situations, phases, capabilities, and properties.

Acknowledgments. This study was financed in part by CNPq (443130/2023-0, 313412/2023-5) and FAPES (2026-PPX8T).

References

1. Ackoff, R.L.: Towards a system of systems concepts. *Management Science* **17**(11) (1971)
2. Almeida, J.P.A., Guizzardi, G., Falbo, R., Sales, T.P.: gufo: a lightweight implementation of the unified foundational ontology (ufo). URL <http://purl.org/nemo/doc/gufo> (2019)
3. de Almeida Falbo, R.: Sabio: Systematic approach for building ontologies. In: *Proceedings of the 1st JOWO on FOIS 2014*, RJ. CEUR Workshop Proceedings (2014)
4. Bunge, M.: *Treatise on Basic Philosophy. Ontology II: A World of Systems*. Springer Netherlands, Dordrecht, Netherlands (1979)
5. Calhau, R.F., et al.: Exploring system behavior in a system ontology. In: *42nd International Conference on Conceptual Modeling, ER 2023*. CEUR (2023)
6. Calhau, R.F., et al.: A system core ontology for capability emergence modeling. In: *International Conference on EDOC*. pp. 3–20. Springer (2023)
7. Calhau, R.F., et al.: Towards an ontology of type-level phenomena for system modeling. In: *International Conference on RCIS*. pp. 121–139. Springer (2025)
8. Fine, K.: Things and their parts. *Midwest Studies in Philosophy* **23**, 61–74 (1999)
9. Guizzardi, G.: *Ontological Foundations for Structural Conceptual Models*. No. 15 (2005)