The Role of the Systemic View in Foundational Ontologies

Riichiro Mizoguchi[a] and Stefano Borgo[b]

[a] *Japan Advanced Institute of Science and Technology (JAIST), 1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan.*

[b] *Laboratory for Applied Ontology, Institute for Cognitive Sciences and Technologies (ISTC-CNR), Trento, Italy.*

Abstract. We start from an informal definition of system to propose a discussion of the systemic view in foundation ontology. The systemic view, by which we roughly mean the view of an entity as a system, plays an essential role to help practitioners model their domains and problems to solve. In the second part of the paper, we discuss the roles of the systemic view with respect to capturing dynamism, causation, function, fault, and behaviors using some examples. The paper aims to increase awareness of in-depth modeling via foundational ontology by harvesting the authors' experience in different application domains.

Keywords. Systemic view, system, foundational ontology, dynamism, function, fault.

1. Introduction

The utility of upper ontologies is at least twofold. On the one hand, there is their intrinsic capability to advance information interoperability. On the other hand, they are powerful systems for in-depth domain modeling. Although both have attracted considerable attention to date, the authors believe that the value of the latter could be even more substantial and long-lasting. The use of ontology for in-depth domain modeling deserves to be explored more in the future. However, to harvest the advantages of modeling via upper ontologies we need to develop an "engineering attitude" that builds on foundational theories.

The current foundational ontologies concentrate mainly on the fundamental types such as objects, properties, processes, and events. These ontologies are highly neutral with respect to application domains and clearly defend this position. It follows that they provide no direct information nor guidelines on how to help practitioners to model specific domains. This neutrality has become increasingly problematic since the foundational ontology developers are not sufficiently engaged to fill the gap between the (defended) ontology's neutrality and its (desired) application in real cases. This gap is particularly evident when the application domain is inherently dynamic and systemic. Foundational ontologies, being part of the applied ontology research area, must start to give practitioners a foundational viewpoint for grasping the dynamism that knowledge engineers find in their domains. For instance, although the notions of function and behavior in YAMATO [7] are quite powerful, knowledge engineers need a more developed and integrated theory that provides a systematic use of these notions. The systemic view can fill this gap when suitably embedded into foundational ontologies.

It is commonly accepted that foundational ontology can be based on different philosophical positions such as realism and constructivism [2][5][10][18]. Yet, all its supporters claim that it has to remain independent of domain-specific views. We think this latter claim relies on a misconception and should be abandoned. We agree that ontology should be domain independent but observe that there are different ways to be domain independent. We introduce the distinction between domain generality vs. domain neutrality. *Domain generality* is here understood as the suitability of an ontology to be applied to different domains. This generality feature is important to ontology. On the other hand, *domain neutrality* is the lack of domain-driven viewpoint in the ontology, and we claim that this feature jeopardizes

FOUST 2021: 5th Workshop on Foundational Ontology, held at JOWO 2021: Episode VII The Bolzano Summer of Knowledge, September 11-18, 2021, Bolzano, Italy
EMAIL: mizo@jaist.ac.jp (R. Mizoguchi); stefano.borgo@cnr.it (S. Borgo)

ontology relevance in applications. When domain independence is understood as domain neutrality, ontology loses its appeal to practitioners.

To emphasize this point, in this paper we present in ontological terms the systemic view, that is, a modeling perspective that can guide domain experts to build models of their domains with engineering mind under the umbrella of a foundational ontology. Engineers follow special interests and views when they build their models. For instance, they often concentrate on functionalities, because a physical artifact is relevant to them in so far as it manifests its designed functions. They also tend to build their models as systems of parts and components because that is how designed systems are conceptualized. However, not all physical entities are functional nor built as designed systems. This engineering viewpoint is driven by their view of the domain. Furthermore, not all actual happenings (term that we use as synonym of events) in a domain are explicitly coordinated in the model. The systemic view is needed to enable domain experts to see all the needed entities as functional entities to provide an ontologically sound and functionally unified view, and to capture relevant happenings into a single system. In this way, they can talk about their views and problems in a coherent manner in terms of functional systems. For these reasons, in this paper we discuss how to view an entity as functional entity and start to develop such a system that can capture different kinds of happenings (expected and coordinated or else).

This paper is organized as follows. The next section presents preliminary ideas for the discussion and proposes an informal definition of a system that serves as a first foundation of the systemic view. In particular, the contingency and the system boundary are discussed as key factors. Section 3 presents the role of the systemic view presented in section 2 and discusses dynamism, causation, functions, fault, device ontology, and behavior. Related work is presented in section 4 followed by concluding remarks.

2. What is a System?

Systems are complex entities that form multiple networks interrelated by structural, functional, processual and behavioral relationships. This complexity naturally leads to different views which are summarized as follows:

(A) Wholistic view

Intuitively, the systemic view helps practitioners capture in a unified view happenings that are otherwise scattered and seemingly unrelated. The purpose is to identify and integrate in a single view those happenings as collaborative and coherent changes of reality and to understand them as achieving a goal which can be attributed to the system itself. Thus, the systemic view provides a wholistic view of happenings. Note that the goal is not always apparent or explicit before introducing the systemic view. In many cases, it is implicit or hidden and becomes clear only after introducing the system by adopting the systemic view. This is particularly important when applied to entities that are better modeled as subsystems, in our experience these entities are often overlooked. The proper understanding of how the systemic view works needs coherent analysis of components' behaviors. This analysis can be successfully performed by appropriate identification of (sub) systems, each of which provides systemic goals that reveal the goal (subgoal) hierarchy embedded in the whole system.

(B) Static behavior

Dynamism is an important feature but not the only characteristic of systems. The systemic view takes care of static as well as dynamic behaviors. Any physical object (unitary and non-dissective continuant) can be seen as built as a system of parts. In many cases, those parts collaboratively contribute to maintaining the structure of the whole by exerting static forces. What is remarkable in such static characterization is that the goal that the collaborative behaviors achieve must be seen as an internal goal rather than external one. That is, keeping the structure of the whole is part of the internal world of the system.

(C) Functional object

Most artifacts in application domains can be successfully analyzed from the functional viewpoint. Indeed, the functional perspective is pervasive and this pushes us to take it at the center of our analysis. Functionality here does not mean that the object's reason of existence is its functional capability. In other words, the object does not need to be intrinsically functional. It is the systemic view that allows to view an arbitrary entity as a functional entity. Imagine a mountain, a material object which is not intrinsically functional. When we look at a flow of air and at how it is impacted by the mountain, we naturally take the view that the mountain performs a blocking function: the mountain prevents the wind from moving across the regions divided by the mountain[1]. When we focus on the rocky cliff of the mountain, we naturally take the view that the mountain provides a place to practice rock-climbing, which is analogous to that a conference room allows people to hold a meeting by providing users with a meeting space. In engineering such a function is called "Allow-type" function and is separated from the functions of the "Achieve-type" [26].

In sum, any physical object can be viewed as a functional object by selecting an appropriate context and behavior, and by setting a related goal. Even a rock on a table admits this functional reading since, e.g., the rock performs a damaging function by constantly exerting a downward force on the table. Such functions are not **intrinsic** but **extrinsic** functions imposed by the users contingently. As we discuss later, closed systems, i.e., systems that do not have interactions with the outer world, are not suitable for functional modeling.

2.1 Key concepts

We now recall four key concepts needed to understand, identify and model systems [8][21].

(1) Goal

A state (technically characterized by a state description) to be achieved. In the literature (e.g., in BDI [28]), goals depend on an (implicit or explicit) intentional agent. In the analysis of a system the goal is isolated to characterize the system. This means that the goal is intrinsic to the system: it is an essential part of it as the goal contributes to defining the system itself. Still, the goal is not always intentional as the system itself may have no intentionality. System's goals can be divided into two kinds: **external goal** and **internal goal**. The former is given by the outer world. In the case of artifacts, they are given by designers and/or manufacturers. In the case of non-artifacts, they are given contingently by the context. The external goal is achieved by the selected behavior of the system, and all the related subsystems (components) collaboratively realize the selected behavior, in which **the realization is the internal goal**, and the goal is decomposed into subgoals each of which is given to each subsystem [21].

(2) Behavior selection[2]

A system usually displays multiple behaviors, i.e., relationships with the external environment and/or across its internal components. In order to explain a system's behavior, one has to choose the interactions that determine it.

(3) Systemic goal

When the goal and behavior of a (complex) entity have been selected, we have a perspective on the entity as a system. Some of its components (subsystems) contribute to the realization of the selected behavior. These components form a network of interactions which collaboratively manifest that behavior of the system. The behavior achieves a state taken as a goal for the system *relatively to the given perspective*. As said, such a goal may not (it does not need to) be intentional. We call the realization of the selected behavior a "systemic goal" because it is specified thanks to the fact that the participating components (subsystems) form a system. (Note that the concept of systemic goal does not depend on how system is defined, so there is no circularity in this notion.)

(4) Function

A function is defined as a role played by a behavior in the systemic context. Such a function is called systemic function and subsumes both artifactual and biological functions [8][21]. The systemic function is intention-neutral and hence it allows us to deal with natural systems (including contingently isolated systems) as functional systems.

[1] Note that it is not unusual. A screwdriver is sometimes used as a hammer where the hammering function is imposed on it.

[2] By behavior, we here mean B1 behavior discussed in 3.C).

We find all the above concepts necessary to define the concept of system. Note that the use of the term 'system' in the introduction of these core concepts is informal and essentially stands for an entity made of multiple components. The characterization of system as an ontological concept will be done later and at that point it will be possible to reread the above notions in more formal terms.

2.2 Informal Definition

`System' is a term used across all domains. Even when explicitly studying systems as entities intertwined with functional and behavioral aspects, as done in this paper, misunderstandings can arise. In this section we collect some common views of what a system is to be able to compare them with our proposal.

I. A generic system as a Structure (See 2.3) is an entity such that:
   (i) is a whole (physical) entity with a boundary separating the world in: the system's inside and the system's outside (the latter can be possibly empty if the system is the universe itself).
   (ii) has one or more components (the simplest case is, e.g., a pebble as a paper weight)
   (iii) if it has more than one component, each of its components must interact with at least another component. More precisely, a multi-component system cannot be divided in two non-interacting subsystems.
   (iv) if it has more than one component, it can be decomposed into multiple subsystems.
   (v) if it has more than one component, the systemic goal of a subsystem is specified by its smallest super subsystem. Intuitively, it is a (possibly non-intentional) goal necessary for sense-making the behavior of a subsystem. As described above, it is automatically specified when the behavior of the system is selected because all the components are supposed to realize the behavior.
   (vi) has input object(s) which is processed and then output.[3] By being processed, we mean: a quality/state of the input object is changed while it goes through the subsystem. The completion of the process with the output release[4] achieves the systemic goal. The detail of the process is defined in the Device ontology discussed in section 3.B).
   (vii) its connected components exchange information or objects as input/output.
   (viii) its subsystems are also systems.

II. An open system as a Generic system is an entity such that:
   (i) has an interaction with the outer world, and hence, there can be input and output flow.
   (ii) its input object is processed by the system and then it is output.
   (iii) is functional only when a goal from the outer world (outside the system) is fixed.

III. A closed system as a Generic system is an entity that has no interaction with the outer world. Such a system has neither input nor output. The system behavior of such a system would be growth, being in an equilibrium/homeostatic state which is an internal goal, rather than a goal of the system itself. Example: The universe, which is the biggest system in reality, is a system with no outside, and hence no input or output. Closed systems are out of the scope of this paper.

IV. An artificial system [6] as an open system is an entity that:
   (i) is the result of an intentional selection of material and the attribution of capabilities [4].
   (ii) may be obtained by manufacturers following instructions by designers with the intention of performing some functions aimed to achieve some goal (technical and engineering artifacts [6]).
   Examples: A **paper weight** is a system with one component (itself). An agent selected it out of several other pebbles and attributed it the capability to hold papers firm without damaging them (the

[3] There can be exceptional cases. Imagine a part-replacement system that receives a new part and output the old part. It performs a kind of a meta-performance that operates on part/component rather than objects on which parts operate. In such a case, a different view must be incorporated to model such a system. That is, the input objects must be components, and hence the "proper" change identified must be understood in terms of the type of the component playing the same role rather than the objects on which components operate. It is apparent from the fact that replacement of a nut with a bolt does not make sense. In sum, I.(vi) should be understood as "a player of the same type of role is processed by the system and it is output."

[4] Precisely, not the output but B1 behavior presented in C) of section 3.

external goal). The system has gravity as the input and weight (the downward force at the bottom) as the output.

A **spinning top** is another one-component system. This system has no external goal in itself. Considering the effect on the observers, however, the systemic goal of the spinning top is to attract observers' attention by keeping standing despite the apparent impossibility.

V. A natural system *as an* Open system is an entity such that:
   (i) has an outer world if it is not the universe.
   (ii) has neither designer nor builder.
   (iii) has a systemic goal to achieve.
   (iv) its boundary, and hence all the components, as well as the systemic goal are selected/identified only for the reason of explanation of the system.

VI. A contingent system *as an* Open system is an entity such that:
   (i) has been born without design and intentional building process.
   (ii) its goal is given contingently (see the cases shown below and in Fig. 3).
   (iii) its components are present as objects before the identification of the system itself and of its boundary. The system arises from the identification of the systemic goal and the system boundary.
   (iv) when the systemic goal is removed, the system disappears, though all the components remain as they are. That is, their behaviors don't depend on the identification (or else) of the system.

For an example of a contingent system, imagine two houses that are built next to each other but apart by, say, 10 meters or so. Case 1. For the layman living in the area or walking along the street the two houses do not form a system because they are independent of each other and have no systemic behavior in daily life. Case 2. Assume now that someone plans to build another house along the street. Among the things considered, there is the air flows around there. If the two houses already present are aligned north-south and the town is notorious for the strong north wind, the new house can be planned south of the other two to take advantage of the disruption of the wind that the two existing houses provide. Indeed, the expectation is that these two houses work as a barrier against the north wind. Thus, in the context of the third house construction, the two existing houses work as a two-component barrier system for the third one. Suddenly, the two houses form a system that weakens the north wind in the area. A possible counterargument would be: The system has not disappeared because those two houses behave exactly in the same way to weaken the north wind independently of the building plan of the third house. However, the claim needs to consider the two houses as a system with respect to the wind and this choice determines the existence of the system and its boundary. The point here is that there is a choice to look at the effect of the two houses together on the wind (the systemic goal), this choice can be determined by the building plan of the third house or by the desire to give a counterargument to that position. In both cases, the system is contingently formed.

As we see in this example, the notion of system we are developing is essentially functional, and indeed it is driven by a theory of artifacts and their purpose. Behaviors are included to realize the required function. Function determines the identity of an artifact, and the artifact's functional structure explains how it works. This said, some people might be skeptical about the functional view of systems because it seems to exclude non-functional systems. Note here that every open system has a behavior, and if the fixed goal is achieved by that behavior, a function is attributed to the system according to the notion of systemic function [21]. We thus can conclude that all open systems are functional. Closed systems are not.

2.3 Ontological Status of System

In this section we present our proposal for an ontological theory of systems that can be integrated in several existing foundational ontologies. We take a system to be an entity that satisfies the "system *is-a* structure" characterization in Section 2.2. The guiding intuition is that structure plays a role similar to that of material in defining a physical entity. This suggests taking the multiplicative approach via the constitution relationship [27], to separate the physical and the systemic view of an entity. The systemic entity relies on the physical entity as constituent. This specific issue needs to be developed formally and is left as future work. Here it suffices to note that the structural hierarchy typical of systems should not mix up with the ***is-a*** hierarchy of ontological entities: the first has the characteristics of a mereological classification (the components and subsystems are parts of the system). Furthermore, the systemic view is not ontologically neutral in the sense that, as seen before, it depends on a few choices: a goal for the entity, a behavior of the entity, a systemic goal for the entity's components, and a function for the identified behavior(s). If we take these elements as identifying a sort of context, the systemic view can be understood as an emerging characterization of physical entities which depends on the chosen context. The constitution relationship between physical objects and systems is also influenced by the inner structure of goals and behaviors. This dependency puts the systemic view as an orthogonal hierarchy with respect to the taxonomical organization of a foundational ontology: the systemic view is better understood as an emerging extension of the ontology that breaks domain neutrality without jeopardizing domain generality. Briefly put, the systemic view enriches foundational ontologies with a context-dependent hierarchy.

3. The Roles of the Systemic View

In the context of in-depth modeling of the real-world and of real-world problem solving, the underlying policy and the types that organize the structure of the foundational ontology are appreciated but also expected to be compliant with the needs to model systems, i.e., in our opinion, with the systemic view.

A) Wholistic models of happenings' dynamics

Most of the existing foundational ontologies mainly provide users with upper-level types which are not enough to help practitioners capture the dynamism of the real world occurrents. An orthodox way to improve this is to develop a (perhaps partial) domain ontology as an extension of a foundational ontology [25]. This works well because such a domain ontology is perfectly compliant with the foundational ontology and, at the same time, helps domain experts to formulate domain concepts and activities, like the manufacturing processes and their resources formalized in [25]. To move beyond the modeling of activities, we need to help practitioners to model their view, which we claim to be mostly a systemic view. This view enables them to capture the dynamism of the domain in a wholistic manner and to see happenings as interconnected events bound by causal relationships. There is a need to model everything within a comprehensive systemic view to successfully capture the different happenings as forming a whole.

B) The device ontology [20] is a conceptual model of open systems and works as a role-assignment system. The central roles are: Device, Conduit, Operand, and Medium. All of them are played by existing entities. Modeling via the device ontology is granularity-free and can generate nested structures.

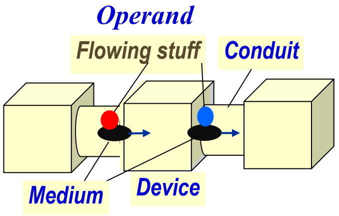


Fig. 1 Device ontology.

The starting point can be stated as follows: a device processes a certain input to produce a certain output. On the device-centered view, an entity is regarded as a composition of devices having as ultimate output the fulfilment of some need on the connecting device. On the process-centered view, in contrast, the entity is viewed in terms of the collection of processes that occur while creating this output. The elements of the device ontology can be viewed also as agents since they play an active role in bringing about a certain output. The process ontology, in contrast, recognizes not agents but participants, entities which participate in the given phenomena as they occur but without playing any agentive role. One can informally understand the difference between these two views considering the networks they generate. A device ontology takes devices as the nodes and the flows of objects from a device to another as links. A process ontology takes objects as the nodes and processes on the objects as links. The two views are complementary to each other.

As said, the device ontology consists of four major roles: **Device, Conduit, Operand,** and **Medium** (See Fig. 1). The Device role is played by an entity when processing an object as input and returning it

as an output (e.g., water heater). The latter object in this process plays the Operand role (the water temperature). The operand is carried to and from the Device by an entity which, doing this, plays the Medium role (the water). Devices are connected between them via entities that play the Conduit role (the pipes connecting the heater) which are a type of device affecting only the location of entities (because of these they are called semi devices). Conduits allow Medium and hence Operand to flow between devices. In this view, there is no representation of the process going on in the device: the device ontology adopts a black box principle so that what is happening inside the device is not modeled. To do so, one has to employ a finer-granular by looking at the device itself as a system and model its parts as devices, conduits, etc. Thus, the device view can be applied repeatedly to form a nested model.
The **B1** behavior (Fig. 2, left) is realized by the process internal to the device, this process is unknown at the level of granularity that the device ontology gives us. The **B0** behavior (Fig. 2, left) is not a causal relationship and can be ignored here. The **B2** behavior (Fig. 2, right) corresponds to the standard notion of causation. Let us focus on behavior now.

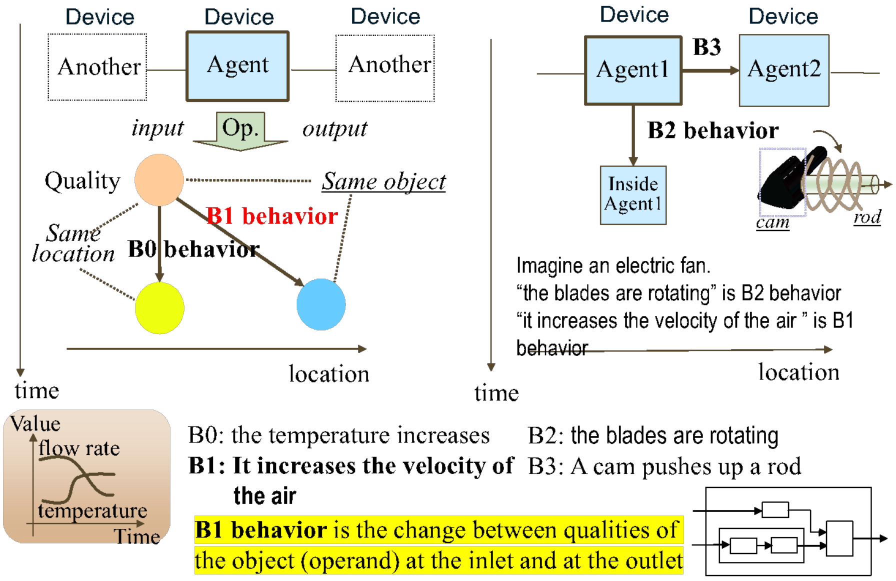

Device
Device
Device
Another
Agent
Another
input
Op.
output
Quality
Same object
B1 behavior
Same location
B0 behavior
location
time
Device
Device
B3
Agent1
Agent2
B2 behavior
Inside Agent1
cam
rod
Imagine an electric fan.
"the blades are rotating" is B2 behavior
"it increases the velocity of the air " is B1 behavior
location
time
Value
flow rate
temperature
Time
B0: the temperature increases
B1: It increases the velocity of the air
B2: the blades are rotating
B3: A cam pushes up a rod
B1 behavior is the change between qualities of the object (operand) at the inlet and at the outlet

Fig. 2 Device ontology and four kinds of behaviors.

C) Behaviors

Behavior is a type that subsumes actions performed by humans, organisms or artifacts. As already anticipated in part, there are four types of behavior in the device ontology: B0, B1, B2, and B3. The behavior mostly studied is about the change in the operand occurring between the input and the output positions (Fig. 2, left): the **B1 behavior**. When we focus on what is happening inside the device we have another type of behavior, called **B2 behavior**. When a device pushes another device to which it is connected, a new type of behavior arises. This behavior is given by the direct interaction between devices and is called **B3 behavior.** The fourth kind of behavior, called B0, is used in numerical simulation. A typical example would be a changing process of the temperature of the fluid measured at, say, the inlet of a device. We call such behavior **B0 behavior**. This latter behavior captures what engineers intend when they say that a set of measurements describe the behavior of the system under consideration. The point here is that B0 behavior does not address causation relationships, instead all the other three behaviors do. As said, B1 behavior does not describe how the change of the operand is realized, it just records the change. This is one of the most crucial characteristics of B1 behavior that represents what change is achieved by the device leaving how it is achieved unexplained. This is the heart of the black box principle. As discussed in [14][15] and [21], B1 behavior plays a significant role in the definition of function. In other words, the identification of B1 behavior and excluding behavior of all the other three kinds is the success factor of the unified definition of function [21].

D) Causality/causation

Causality manifests its engineering implications via the systemic view. Fig. 3 gives an example of causation in the systemic view and of the relationships this view can highlight and help model. In the example Tom throws a stone to a window which breaks because of the hit. We look at this case as an emerging system, a preliminary step to discuss causation. According to our proposal, before the stone-throwing event, there is no system in the scenario since there is no functionality at play. Tom, the stone and the window exist independently of each other and do not form a system. A system emerges when Tom throws the stone to the window since this activity connects all these entities with a causal relationship "has effect": Cause process/event → Effect state/process (C → E, for short)[5] [22][26]. The identification of a causal relationship determines the existence and extent of the system. In other words, causation reveals a certain connection between the participants of the related occurrents and this connection is an essential feature for a system to exist. At the time when Tom holds a stone and starts to throw it, Tom (or better Tom's arm), the stone and the window virtually form a system (a functional device) in which three behaviors (occurrents) happen in turn: throwing motion, stone traveling, and stone hitting the window. Whether or not Tom has an intention to break the window (e.g., whether it aims at hitting the window) does not matter in this view. Once a causal relationship "Tom throws a stone to the window and as a result the window breaks" is picked out, we can talk about a window-breaking system composed of Tom and the stone together with the window as the affected object.

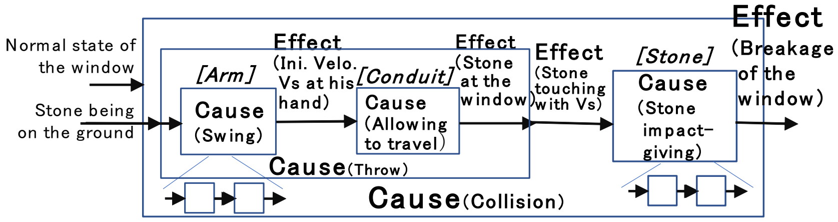

Normal state of the window
Stone being on the ground
[Arm]
Cause (Swing)
Effect (Ini. Velo. Vs at his hand)
[Conduit]
Cause (Allowing to travel)
Effect (Stone at the window)
Cause(Throw)
Effect (Stone touching with Vs)
[Stone]
Cause (Stone impact-giving)
Effect (Breakage of the window)
Cause(Collision)

Fig. 3 Configuration of the window-breaking system, in which all input and output arrows indicate states, and all boxes labeled "Cause" indicate events. In the finer granularity, some causal relationships are of type: Process => Process (e.g., arm-swinging => stone-flying). The releasing event is omitted (adapted from [22]).

The system can be analyzed at three-levels of granularity (See Fig. 3). The coarsest granularity suggests modeling the phenomena as a system (device) composed of the stone and Tom's arm whose input is the window (in a certain state) and output is the window (in a different state, i.e., with the glass broken): the window is the medium and its state is the operand, and the device structure is hidden so that Tom, the stone, and the conduit are invisible. This coarsest device (the whole system) models that something happened in side the system and the window has been broken as a result. All the three occurrents contribute to the overall behavior of the system. What happens within this system is analyzed at the second level of granularity. At this level, two subsystems are identified: (1) the stone as a window-hitting subsystem and (2) Tom's arm and an immaterial conduit (the 3D region occupied by the trajectory of the stone) with the stone as a stone-travelling subsystem. In the first subsystem, the stone hitting the window with a certain velocity corresponds to the cause, and the breakage of the window to the effect. Here the medium is the window whose state is the operand, and the device is the stone. The second subsystem, which corresponds to the throwing action, models the stone moves from Tom's hand to the window with a certain velocity. In this subsystem the operand is the stone's location and velocity, the medium is the stone, and the device is composed of Tom's arm and the conduit (though invisible). At the third level (inside of the second subsystem), the swing action and the stone traveling, that is, how the stone reaches the window are modelled: as to the swing action, the operand is the stone's velocity, the medium is the stone, the device is Tom's arm, and as to the traveling process, the conduit is the device that allows the stone whose location is the operand to move from Tom's hand to the window. Note that the conduit appears finally here at the third level and the stone plays two roles: device and

[5] More generally, C and E stand for any occurrent including state, process, and event.

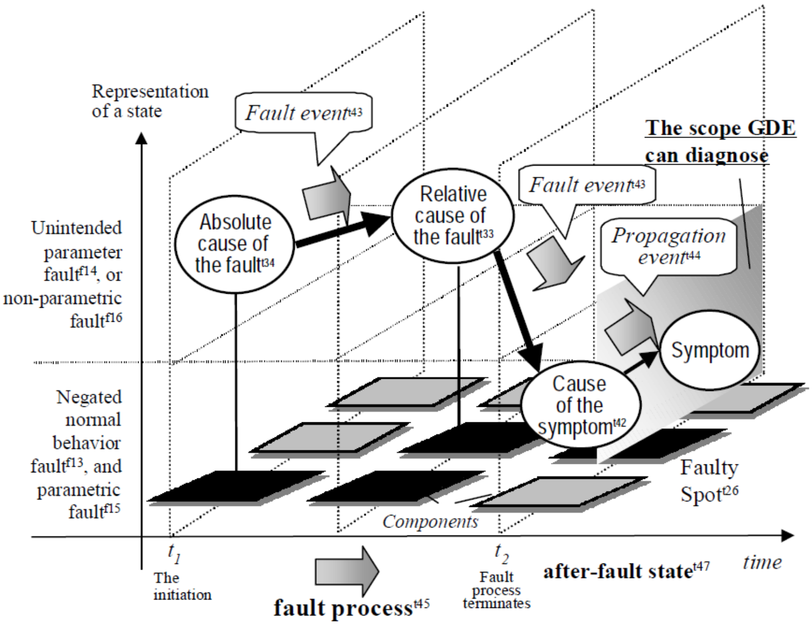
medium (no operand role). The resulting configuration of system/subsystems is shown in Fig. 3 where one can see how the three granularity-views combine.

This analysis can be turned into a general rule for modeling causality using the systemic view and the device ontology.

Modeling causality via the systemic view: Identify the participants of the occurrent, fix the goal and the emerging system, apply the device ontology analysis to assign E as output (what to achieve) and C as internal behavior (how to achieve). If necessary, repeat recursively to C (e.g., in the case where C is a causal chain made of finer-granular occurrents).

E) Functions

A function is a teleological entity, and it needs a goal. From our discussion on the systemic view, a function is always a function of a system [14][15][21]. Function is thus intrinsically systemic. In other words, an isolated system cannot perform any function. Any of its components can perform a function, e.g., to achieve the internal goal set by the system as a whole (this is the only way we can talk of function in the universe as system). However, the function of the system, which is what we are talking about, cannot exist if the system is isolated.



Representation of a state
Fault event^t43
The scope GDE can diagnose
Unintended parameter fault^f14, or non-parametric fault^f16
Absolute cause of the fault^t34
Relative cause of the fault^t33
Fault event^t43
Propagation event^t44
Symptom
Cause of the symptom^t42
Negated normal behavior fault^f13, and parametric fault^f15
Faulty Spot^t26
Components
t1
The initiation
fault process^t45
t2
Fault process terminates
after-fault state^t47
time

Fig. 4 Fault processes and the systemic view (Adapted from [13]).

F) Faults

A fault indicates that a component of a system doesn't properly function, i.e., it does not manifest a suitable behavior. An ontology of faults is given in [13]. The General Diagnostic Engine (GDE) proposed by de Kleer [16] can efficiently find a faulty component of a system from given symptoms using the expected behaviors of every component in the system based on the Assumption-based Truth Maintenance System (ATMS) [17]. GDE is ontologically limited, it can find only a component that is responsible for explaining the malfunctioning symptoms. GDE cannot identify why the found component is faulty from the theoretical viewpoint. GDE relies on the system structure and the expected behavior of each component. This means that the system is fixed from the start, it is a view of the system as it was intended, not as it is. In other words, there can be unintended structures which usually remain hidden. Also, a malfunctioning, rather than a fault component, occurs usually in unintended systems. For example, imagine a motor for generating rotation energy: it should turn the shaft with the motor, where the shaft forms an intended subsystem. While running, the motor generates heat energy which propagates to components located around it. If a component's performance is affected by temperature, an unintended connection emerges between the motor and the component: this is an unintended subsystem. GDE detects the failure of a component but not the subsystem that causes the failure, thus not the component or relationship which is at the origin of the failure (Fig. 4). The heat propagation from the motor to the component corresponds to Tom throwing the stone to the window in Fig. 3 (see [13] for details), these (unintentional, undesigned) systems play a crucial role in fault analysis.

4. Related Work

In addition to information interoperability and in-depth modeling of domains, foundational ontologies have been used also for conceptual modeling [11] which requires a solid foundation of upper types. As discussed earlier, there are standard ways to extend foundational ontologies to model application domains, examples are [25] in which manufacturing resources are ontologized and [23] focusing on surgical processes. Systems are discussed also in information science [12].

As to the definition of systems, there are many proposals [19][24]. Some typical examples are:

"A system can be defined as a set of elements standing in interrelations. Interrelation means that elements, p, stand in relations, R, so that the behavior of an element p in R is different from its behavior in another relation, R'. If the behaviors in R and R' are not different, there is no interaction, and the elements behave independently with respect to the relations R and R'." [2]

"A system is a set of two or more elements that satisfies the following three conditions. (1) The behavior of each element has an effect on the behavior of the whole. (2) The behavior of the elements and their effects on the whole are interdependent. [...]" [1]

Although these definitions are well motivated and capture some important points, they model the complexity of systems only in part and are unsuitable to introduce an ontological view. On the other hand, our definition is made from the ontological viewpoint incorporating key concepts such as the system boundary, the systemic goal, and contingency with fundamental classification of systems.

A criticism of Perez [24] directed to Bertalanffy's definition is applicable to our definition as well. Perez claims that Bertalanffy's definition is useless because it includes all the physical entities. We understand his concern. However, we believe that this is needed. In fact, any physical entity is a system of parts when viewed from the structural point of view as suggested in section 2.4. A table is an amount of wood from the material point of view, and is a system of several parts (e.g., a table top and some legs) from the structural point of view.

5. Concluding Remarks

We have introduced and discussed the systemic view, what is its role in foundational ontologies, and how it can help practitioners cope with hard problem solving in application domains. The proposed definition of system is more elaborated than existing ones and contributes to the talk about roles in the systemic views. In particular, a unified framework for causation/causality, functionality, and fault analysis becomes possible when introducing the systemic view. In the future we will develop a formalization of the definition of system, the related notions and the classification of systems that it generates which can cover many different views [9].

References

[1] Ackoff, R. L., 1981, Creating the Corporate Future, Wiley, New York

[2] Arp, R., Smith, B. and Spear, A. D. (2015). Building Ontologies with Basic Formal Ontology, MIT Press.

[3] Bertalanffy, L. von, 1969, General Systems Theory, Braziller, New York.

[4] S. Borgo, L. Vieu, Artefacts in formal ontology, Philosophy of technology and engineering sciences, 273-307, 2009.

[5] Borgo, S. and Masolo, C. (2010). Ontological Foundations of DOLCE. In R. Poli, M. Healy and A. Kameas (eds.), Theory and Applications of Ontology: Computer Applications (pp. 279–295), Springer.

[6] Borgo, S., Franssen, M., Garbacz, P., Kitamura, Y., Mizoguchi, R. and Vermaas, P. E. (2014). Technical artifacts: An integrated perspective, Applied Ontology 9 (3–4), 217–235.

[7] Borgo, S. and Mizoguchi R. (2014). A First-order Formalization of Event, Object, Process and Role in YAMATO, In P. Garbacz and O. Kutz (eds.), Proceedings of the 8th International Conference on Formal Ontology in Information Systems (FOIS2014), Rio de Janeiro, Brazil, September 22-26, 79–92, 2014.

[8] S. Borgo, R Mizoguchi, Y Kitamura, Formalizing and adapting a general function module for foundational ontologies, Formal Ontology in Information Systems (FOIS2016), 241-254.

[9] S. Borgo. "An ontological view of components and interactions in behaviorally adaptive systems." Journal of Integrated Design and Process Science 23.1 (2019): 17-35.

[10] N.B. Cocchiarella. Logic and Ontology. *Axiomathes* **12,** 117–150 (2001). https://doi.org/10.1023/A:1012758003706

[11] Guizzardi, G. and Wagner, G. (2010). Using the Unified Foundational Ontology (UFO) as a Foundation for General Conceptual Modeling Languages. In R. Poli, M. J. Healy and A. D. Kameas (eds.), Theory and Application of Ontology: Computer Applications (pp. 175–196), Springer.

[12] Ahmad Kayed, Robert M. Colomb, Using BWW model to evaluate building ontologies in CGs formalism, Information Systems, 30 (5), 379-398, 2005.

[13] Y. Kitamura and R. Mizoguchi: An Ontological Analysis of Fault Process and Category of Faults, Proc. of Tenth International Workshop on Principles of Diagnosis (DX-99), pp.118-128, June 8-11 1999.

[14] Y Kitamura, M Kashiwase, M Fuse, R Mizoguchi, Deployment of an ontological framework of functional design knowledge, Advanced Engineering Informatics 18 (2), 115-127, 2004.

[15] Y Kitamura, R Mizoguchi, Characterizing functions based on phase-and evolution-oriented models, Applied Ontology 8 (2), 73-94, 2013.

[16] de Kleer J., and Williams, B. C., Diagnosing multiple faults, Artificial Intelligence, 32: 97-130,1987.

[17] Johan de Kleer: An Assumption-based TMS, AI Journal, 28 (2), pp.127-162 (1986).

[18] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, and A. Oltramari. WonderWeb Deliverable D18. Technical report, Laboratory for Applied Ontology ISTC-CNR, 2003.

[19] Miettinen, K. (2018). The Ontology of Systems. Ontology for the Intelligence Community 2018, Monterey. https://tinyurl.com/y8mhnooa.

[20] R Mizoguchi, Y Kitamura, A functional ontology of artifacts, The Monist 92 (3), 387-402, 2009.

[21] R Mizoguchi, Y Kitamura, S Borgo, A unifying definition for artifact and biological functions
Applied Ontology 11 (2), 129-154, 2016.

[22] R Mizoguchi, Causation: revisited. Proc. of the Workshop, JOWO, 2020.

[23] D Neumuth, F Loebe, H Herre, T Neumuth. Modeling surgical processes: A four-level translational approach. Artificial intelligence in medicine 51 (3), 147-161, 2011.

[24] Gustavo de Jesus Perez. On the Essence and Ontology of Systems, Open Science Journal, Vol 5, No 3 (2020).

[25] E.M. Sanfilippo, W. Terkaj, S. Borgo, Ontological modeling of manufacturing resources, Applied Ontology, 1-23, 2021.

[26] Toyoshima, F., Mizoguchi, R. and Ikeda, M. (2019). Causation: A functional perspective. Applied Ontology, 14(1), 43-78.

[27] L Vieu, S Borgo, C Masolo, Artefacts and roles: Modelling strategies in a multiplicative ontology
Formal Ontology in Information Systems, 121-134, 2008.

[28] E. Yu, Modeling Strategic Relationships for Process Reengineering. Social Modeling for Requirements Engineering 11.2011 (2011): 66-87.