Data & Knowledge Engineering 141 (2022) 102062

Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

System: A core conceptual modeling construct for capturing complexity

Roman Lukyanenko [a,*], Veda C. Storey [b], Oscar Pastor [c]

[a] University of Virginia, United States of America
[b] Georgia State University, United States of America
[c] Universitat Politècnica de València, Spain

ARTICLE INFO

Keywords:
System
Systemism
Conceptual modeling
Complexity
CESM+
Emergent properties
Ontology
Bunge Systemist Ontology (BSO)
Retrospective case study
Citizen science

ABSTRACT

The digitalization of human society continues at a relentless rate. However, to develop modern information technologies, the increasing complexity of the real-world must be modeled, suggesting the general need to reconsider how to carry out conceptual modeling. This research proposes that the often-overlooked notion of "system" should be a separate, and core, conceptual modeling construct and argues for incorporating it and related concepts, such as emergence, into existing approaches to conceptual modeling. The work conducts a synthesis of the ontology of systems and general systems theory. These modeling foundations are then used to propose a CESM+ template for conducing systems-grounded conceptual modeling. Several new conceptual modeling notations are introduced. The systemist modeling is then applied to a case study on the development of a citizen science platform. The case demonstrates the potential contributions of the systemist approach and identifies specific implications of explicit modeling with systems for theory and practice. The paper provides recommendations for how to incorporate systems into existing projects and suggests fruitful opportunities for future conceptual modeling research.

1. Introduction

With continued human development, social, economic, political and technological systems are growing more complex [1–4]. Complexity in systems refers to the number of component-parts along with the way in which these parts are structured and interact with one another and with other systems [5,6]. Systems are the complex entities which constitute the world, such as atoms, animals, airplanes, universities, stock markets, and galaxies. Generally, the more complex the system, the more difficult it is to fully predict its behavior. To create and effectively manage complex systems, improved methods, machinery and knowledge are necessary. This "complexity challenge" opens new opportunities for information technology (IT) development to support, create and manage complex systems and their users.

Conceptual modeling is a phase of information technology (IT) development. It traditionally focuses on capturing user requirements, facts and beliefs about an application domain [7–10]. Since the 1970s, database design, especially in large organizations, relied on conceptual models – the products of conceptual modeling – to model the data to be stored in relational databases [11–13]. Another important application of conceptual models is to support business process management and engineering [14–16]. More broadly, conceptual models are used to improve domain understanding, to facilitate communication among IT developers and stakeholders, and to help visualize and solve IT design challenges [10,17–20]. Our growing reliance on information technologies and their increased sophistication necessitates an ever greater ability of conceptual modeling to represent both physical (including mental and social) and digital systems [21].

* Corresponding author.
E-mail address: romanl@virginia.edu (R. Lukyanenko).

https://doi.org/10.1016/j.datak.2022.102062
Received 15 August 2021; Received in revised form 2 August 2022; Accepted 3 August 2022
Available online 23 August 2022

To appreciate the challenge in creating and managing complex systems, consider one of the unrealized solutions for tackling the COVID-19 pandemic, namely, the development of a social (physical) distancing app. Such an app would sense an approaching person and vibrate, thereby alerting the user of the need to keep distance. The anonymized and aggregated data from such app could be used by governments to support data-driven policies and facilitate smarter pandemic response.[1] If widely practiced, physical distancing leads to significant reductions in respiratory disease transmissions [22–24]. However, an effective physical distancing technology is incredibly challenging, not only because of its many technological obstacles, but importantly, because of a host of social, ethical, legal, medical, and psychological challenges [25]. Precise and accurate conceptual modeling of facts and opinions in this domain could assist in the development of effective solutions. Since distancing technology involves personally sensitive usage by millions of people in real time, and, assuming such usage is not mandated, but is voluntary, we need to accurately model values, intentions, motivations and needs of different people in order to align the technology with these needs. The use of such technology is fundamentally collective, involving coordinated efforts on the part of citizens, governments, and medical establishments [26]. To create and sustain an app at such scale amounts to the development of a highly choregraphed complex socio-technical system. Worse still, such a system might behave in potentially unpredictable and even, possibly, dangerous ways. Inadvertently, such app could cause undesirable changes in patterns of human movements and socializing or trigger an expansion of mass surveillance. Measures need to be put in place (including at the level of technical design) to proactively detect and curb any negative outcomes, while promoting the positive ones. Conceptual modeling then becomes a valuable tool to help engineer effective IT solutions to the expanding challenges of humanity.

The objective of this research is to examine existing conceptual modeling capabilities with respect to the challenges of the modern world and suggest a path for better handling of its complexities. We rethink conceptual modeling theory and practice by investigating a thus far overlooked conceptual modeling concept, namely that of "system". Specifically, we propose that the construct of "system" should be regarded as a basic conceptual modeling construct, on par with constructs such as "entity", "attribute", "role", "event" or "relationship".

Amending conceptual modeling languages with the construct, "system", follows a long line of research that introduced additional constructs to increase the expressive power of modeling languages. From early research on conceptual modeling, until present, researchers have been proposing new constructs (e.g., [12,27–32]). Some of these became ingrained in widely used conceptual modeling languages, such as the entity-relationship diagram (ER), Unified Modeling Language (UML), Business Process Model and Notation (BPMN), and Object-role modeling (ORM), which are now staple elements in practice. The sub/super classes (i.e., generalization/specialization relationships) is one such example [33,34]. Similarly, we argue that the construct of "system" has the potential to become another basic and indispensable construct in the world of ever-increasing complexity.

Of course, it is already possible to model system components (e.g., parts of a whole) using conventional approaches, such as ER diagrams or UML. However, as we demonstrate in the paper, traditional conceptual modeling approaches struggle to model many aspects of systems such as *emergence*. Furthermore, even though the notion of system is ubiquitous in the conceptual modeling discourse, there is little guidance for modelers on how to appropriately model systems. This problem is exacerbated by the lack of consensus and clarity on what constitutes a system and its related constructs.

To keep up with the relentless pace of digitalization of business and society, it is important to continue refining conceptual modeling to make it more expressive for cases when more explicit and comprehensive modeling of systems is beneficial. Since these scenarios are pervasive, modeling systems more explicitly is becoming pressing.

In this research, we propose a set of basic notions that are related to the *system* construct, position system as a core conceptual modeling primitive, explain the limitations of existing modeling languages, and outline research initiatives that could further incorporate the system construct into conceptual modeling. Based on theoretical foundations, we propose a CESM+ template for conducing systems-grounded conceptual modeling. Several new conceptual modeling notations are introduced for practitioners and as input into future academic research. The systemist modeling is analyzed in a case study of the development of a citizen science platform. We then provide methodological guidelines for designers and a future conceptual modeling research agenda.

2. Background

2.1. Conceptual modeling constructs

Conceptual modeling research and practice is now over 50 years old, with popular conceptual modeling languages, such as the entity-relationship model [27] appearing in the 1970s. During this lengthy period, numerous constructs have been proposed as hundreds of different conceptual modeling languages and approaches were introduced, evaluated and applied [35–40].

For example, a core modeling construct, which emerged as early as the first conceptual modeling languages, is that of an *entity type* [27]. Entity types or classes (used in entity-relationship diagrams and UML Class Diagrams, respectively), are commonly used to represent groups of objects of interest in the domain of the information systems being developed [13,41,42]. Debates related to these constructs centered on how to appropriately select [41,43] and apply them [44–46], such as identifying the relationship between classes or entity types and the objects they represent [47–51]. Debates also focused on the nature of instances themselves; for example, whether classes can be instances of other classes [52–54].

Other focal constructs in conceptual modeling dealing with entities include "attributes" (characteristics, dimensions, or features of entities), "relationships" (associations among entities) (e.g., [27]), and "roles" (behaviors and functions of entitles) [28,38].

[1] Attempts to develop such technology have been made, but the resulting apps not been widely embraced by society. See, https://spectrum.ieee.org/news-from-around-ieee/the-institute/ieee-products-services/social-distancing-heres-an-app-for-that.

These are common in conceptual models representing the form and structure of domains [18,55]. Modeling approaches representing processes and dynamics of domains include such constructs as "events", "activities", or "gateways" [15,56–58]. Those dealing with goals, values, intentions, have also became popular, having such constructs as "goal", or "actor" [32,59–61].

There have been many debates about the value and limitations of various constructs, as well as proposals for how to use them effectively in conceptual modeling diagrams [62–65]. An overlooked, but extremely important construct is that of "system" and its associated constructs, including emergent properties, mechanism, environment, among others.

2.2. Conceptual modeling foundations and the system construct

The absence of an explicit "system" construct in mainstream conceptual modeling languages (e.g., UML, BPMN, ORM, ER, *i**) is surprising given the ubiquity of the system concept in discourse related to conceptual modeling.

First, systems notion is synonymous with the product of IT — the software or computer applications are widely recognized to be *information systems*. This is well understood in conceptual modeling. As Mayr and Thalheim [8, p. 2] remind us: "from the very beginning, conceptual modeling was propagated as a means to improve the design and implementation of whatsoever *software system*, especially with regard to a comprehensive and as clear as possible elicitation and analysis of *system* requirements". (p. 2; emphasis added).

Second, when IT get implemented in real-world settings, they become part of socio-technical systems [66–68]. *Socio-technical systems* are composed of technical systems (processes, tasks, and technological infrastructure) and social systems (humans, their relationships and social structures). The two systems, when put together and begin to interact, produce joint outputs (e.g., information, furniture, raw materials, services) [69–71]. For example, implemented into organizational settings enterprise resource planning, customer service, electronic payments, e-commerce IT become parts of socio-technical systems created by the fusion of humans and technology. Hence, enterprise diagrams, such as a UML class diagram, BPMN model or ArchiMate diagram [72], commonly model socio-technical systems [2,3,73,74]. Often enterprise models comprise of layers (e.g., [2,73]), which can be understood as systemic levels (discussed later).

Third, the domains that are managed by IT are commonly understood as *systems*. For example, a conceptual model may represent facts about an inventory control system to facilitate a more efficient inventory management by an ERP developed with the help of this conceptual model [75]. Similarly, in an *i** framework, modelers can represent social systems, which contain potential users of technology and their goals and intentions [76].

Technologies, including IT, are also viewed as components of *work systems;* that is, systems in which human participants and/or machines perform work using information, technology, and other resources [77–79]. Similarly, *design* and *use* of information technologies are considered to be ingrained and inseparable from the broader social systems in which they reside [80]. These ideas are accepted in conceptual modeling. Hence, Yu [76, p. 100] explains the benefits of *i** as follows: "unlike traditional systems analysis methods which strive to abstract away from the *people aspects of systems*, i* recognizes the primacy of social actors" (emphasis added).

Fourth, theoretical foundations of conceptual modeling engage with the notion of system. Hence, as we already discussed, work systems theory is positioned as a foundational theory underlying information systems [77–79]. Another theoretical foundation of conceptual modeling is ontology [55,81–85]. Ontology is a branch of philosophy that studies what exists. A popular ontology in conceptual modeling, the Bunge Wand Weber (BWW), contains the notion of system. In BWW, "a set of things is a system if, for any bi-partitioning of the set, coupling exist among things in the two subsets" [86, p. 222]. Some extensions of this ontology, namely Bunge Systemist Ontology (BSO) [87] and Realist Ontology of Digital Objects and Digitalized Systems [88], extend and modify BWW by incorporating additional systems constructs, including emergent properties, mechanism, and levels.

In addition to the ubiquity of systems notion in discourse related to conceptual modeling, as these examples show, there is a great diversity of ideas, approaches, and theories related to systems. Hence, if a given conceptual modeling project were to explicitly adopt a "systemist perspective", it is not clear which approach the modeler should choose and what the basic elements of systems are that need to be represented.

2.3. Conceptual modeling languages and the system construct

Beyond the general presence of systemist notions in conceptual modeling discourse, aspects related to systems are present in conceptual modeling languages themselves. Many conceptual modeling languages, such as UML Class Diagrams or Extended ER diagrams, contain constructs such as "*part of*". These are systemic notions because they deal with composition of complex entities (i.e., systems). Other languages may not contain explicit system constructs but can be interpreted as being systemist. Thus, the dependencies in *i** can be considered emergent properties, which emerge as a result of the interactions among actors.

Some niche modeling languages provide greater support for systems; most notably, the Systems Modeling Language (SysML), a modeling language for systems engineering applications [89,90]. Although it uses the term "system", SysML lacks precise, well-grounded definition of system. The references to "system" are generic and vague. That is, SysML supports the specification, analysis, design, verification, and validation of a broad range of systems, without providing a precise conceptual characterization of what, exactly, a system is. As an extension of a subset of the Unified Modeling Language (UML), SysML inherits the conceptual imprecision of significant concepts. The SysML language's extensions were designed to support systems engineering activities from a generic methodological perspective. Furthermore, SysML does not engage with basic systemic notions, such as emergent properties, except in a very incidental manner (e.g., [90], p. 335).

Overall, the construct of system and its related constructs have been surprisingly underrepresented in conceptual modeling theory and practice, including in popular conceptual modeling approaches [36,37,55,91]. Furthermore, there is considerable ambiguity when referring to systems in IT [92]. Remarkably, if one could ask systems engineering experts for a precise definition of a "system", it is most likely that many different definitions would be provided. Indeed, such is the case among scientists as well [93].

Next, we seek to better understand the notion of system and provide an ontologically supported characterization of the "system" construct and its related constructs.

3. Understanding the nature of systems

The term system is Greek in origin (*systema*), with original meanings of "organized whole, body" as well as "standing together, standing in relation, or togetherness" [92, p. 209]. It may, however, have an even older Sanskrit root, from the cognate word *samsthana*, which also means "being, existence, life" and "standing together".[2]

Once introduced in the 17th century English, the term eventually became an integral part of the vocabulary in philosophy, natural and social sciences, engineering, humanities, and medicine. It acquired an additional sense, subsuming an old saying commonly attributed to Aristotle: "The whole is something over and above its parts, and not just the sum of them all" [94].

3.1. General understanding of systems in science

Today, system is among the basic scientific notions. Indeed, progress in sciences often occurred when what was once considered indivisible (e.g., atom) was later understood to be complex and was conceptualized as systems [95–98]. It is also notable that, in the field of information systems research, which deals with the design, use, and impact of IT on individuals and collectives, there have been repeated calls for *more* systemist theorizing [66,99–101].

System is considered to be a unifying scientific construct [95]. Unfortunately, each discipline, and even sub-disciplines, understand the notion of system in a somewhat unique way, leading to over 100 different definitions and senses of the term [92]. Considering the interdisciplinary nature of a system-based view of reality, general systems theory (GST) was developed by von Bertalanffy [95] and became widely applied [79,102–106]. The basic tenets of GST are as follows. The GST views systems as a grouping of interdependent parts of a common whole. Thus, Ackoff [107, p. 662] defines a system as "an entity which is composed of at least two elements ... each of a system's elements is connected to every other element, directly or indirectly. No subset of elements is unrelated to any other subset". Some systems exhibit emergent behavior. This common whole tends to be resilient to change, or homeostatic, giving systems their stability. In some systems, such as organic or certain artificial systems, support feedback loops exist, wherein the outputs of the system become its inputs, and hence can modulate or amplify the system's behavior. Systems may be closed or open, depending upon whether components of the system may interact with the components of other systems.

3.2. Foundations of ontological systemism

Owing to the adoption and further development of general systems theory and its applications to different scientific domains, many theories and models of systems emerged [6,103,107–111].

We adopt the theoretical lens of *general ontology,* which has been amongst the most important theoretical foundations for conceptual modeling [55,84,112–114]. It is a source of theoretically grounded, consistent, formalized, and rigorous meaning for the basic notions of what exists. Indeed, conceptual models via concepts and their relationships (i.e., constructs) seek to represent facts and beliefs about the world by using constructs assumed to be capable of representing these facts and beliefs.

We use the ontological theory of the philosopher physicist Mario Bunge as our guiding ontological theory. The ideas of Bunge are especially relevant for four reasons. First, they encapsulate the recent advances in sciences, including the debates around the notion of system. Being a scientist himself, Bunge contributed to these debates, publishing his work in physics, chemistry and biology outlets (e.g., [115]). Second, Mario Bunge, mainly via BWW, has been fruitfully used as a reference (or benchmark) in past conceptual modeling research ([2,84,116–119]; cf. [120]). Hence, his ideas have already been considered as relevant for the development of conceptual modeling. Third, Bunge's approach to systems is a general one, mostly compatible with the views held by other proponents of systems thinking and of GST.

Finally, Bunge's objective was the development of a consistent, formalized, and rigorous ontology of systems. This is very important considering the many disagreements and debates surrounding the notion of system. Furthermore, even Bunge was at times inconsistent, owing to the great volume of research and evolution of views [87,121–123]. In this paper, we adapt and extend Bunge's ontology by formalizing it further. In addition, since Bunge did not engage with certain systemist notions which could be germane for conceptual modeling (e.g., systems as *optional* mental abstractions), we synthesize the views of Bunge with select tenets from GST.

Much of familiarity with Bunge in conceptual modeling stems from the BWW ontology developed by Wand and Weber [119,124] based on Bunge's *Treatise on Basic Philosophy* [125,126]. However, this ontology did not offer extensive elaboration of systems and its related constructs (although, in addition to the system construct, the ontology contained the construct of emergent properties). Upon review of Bunge's broader works, Lukyanenko et al. [87] proposed a new ontology, called the *Bunge Systemist Ontology (BSO),* based on the writings of Bunge later in his life [96,121,127]. This ontology, as the name suggests, deals with systems more extensively. However, some germane constructs were still left out (e.g., system level, semiotic systems). Recently, Lukyanenko and Weber [88] developed a *Realist Ontology of Digital Objects and Digitalized Systems* by combining Bunge's ontological notions (including that

[2] https://sanskritdictionary.com/?q=sa%E1%B9%83sth%C4%81na.

of systems, levels and emergent properties) with his theories of semantics and semiotics [128]. To develop the foundations of systems-grounded conceptual modeling, we adopt and extend these, more recent and extended ideas of Bunge.

The BWW ontology postulates that *reality* is made of *things*, which have properties [125, pp. 26–29]. Things are "substantial individuals", which could be *composed* of other individuals or be *simple*, structureless and atomic [118, p. 126]. In his more recent works, Bunge proposed that every object of existence is likely a ***system***. According to the later Bunge, ***the world is made of systems***. Lukyanenko et al. [87] provide three explanations of this, somewhat radical, ontological position.

First, *the notion* of system allowed Bunge to reason about constituents of reality that would be difficult to call *things*. Bunge found that system is more consistent with the scientific and day-to-day discourse. Scientists routinely refer to fields or atoms as "systems", and rarely call them "things". Second, this linguistic practice, for Bunge, reflected deeper ontological reasons. Bunge argued that *there are no simple, structureless entities* — all constituents of reality are complex. Third, as follows from the premise that the world is made of systems, Bunge asserted that ontological systemism provides a more faithful approach for describing reality, and better maps to reality [96,129].

The claims by Bunge have been increasingly supported by modern quantum physics, including the candidate for the unifying theory, the M-theory, which considers fields (e.g., electromagnetic field) as being made of particles, such as *bosons* [130]. Hence, fields, which give rise to the physical forces (e.g., electromagnetism), can indeed be considered systems. The notion of known elementary particles likely being complex is also supported by other quantum theorists [131].

Bunge provides a broad definition of a system as a "complex object every part or component of which is connected with other parts of the same object in such a manner that the whole possesses some features that its components lack – that is, emergent properties" [93, p. 20]. We adopt the same definition for our analysis. In essence, this definition suggests that, for something to be considered a system, it needs to be composed of other components. These components need to be connected to one another and, through these bonds, emergent properties arise.

Bunge defines *emergent properties* as properties of systems which the components lack, and which only appear once the components become part of the whole by interacting with one another [129]. What makes systems especially important and challenging to investigate is that the emergent properties are not directly derivable from the knowledge of the properties of the components. These properties emerge in an organic way once the components become part of the whole. For example, humans individually lack the property "cohesive". This property emerges when the humans form a team or family (both, social systems), and the team is identified as being "cohesive". Similarly, the concept of "commitment" arises as a result of bonding between several people. The same holds true for human-made systems. A house can be "cozy" when furniture is put together in a particular manner. Likewise, music can be considered "soothing" despite individual soundwaves lacking this property.

It should be noted: while all entities per Bunge are complex, they do not necessity bond with *all* other systems to form bigger systems. An electron somewhere on the Moon does not form a system with a bird on Earth. A pile of laptops does not make a socio-technical system with the humans in its vicinity. Still, these systems can be grouped together in a mind, for some purpose. These unrelated groups of systems can be called *aggregates* (or *collections*). For example, we can group together Jupiter's moons, Australian marsupials, Mario Bunge, and the ER (Entity-Relationship) 2022 conference, to make a point about systemist ontology. This group, seemingly unrelated, can have properties in common (such as *located in the Solar system*). However, it is an aggregate, not a system. What distinguishes systems from aggregates is the presence of emergent properties resulting from the systemic interactions among components. Hence, it is not ontologically consistent to treat *Jupiter's moons, Australian marsupials, Mario Bunge,* and *the ER 2022 conference* as a system and thus seek a system-based conceptual modeling construct to show it in a diagram.[3]

In addition to the emergent properties, systems have *hereditary* or *aggregate properties*. These properties are also the properties of the whole, that is the entire system, but they are directly derivable from the properties of the components. For example, *total family income* is a property that is the sum of individual incomes of the family members. Similarly, *mass* of an organism is the sum of masses of all organs and tissues.

Bunge distinguishes two kinds of system: *conceptual* and *concrete* [93, p. 270]. *Concrete* systems are systems made of energy-harboring material components and may undergo change. Concrete systems change in the virtue of energy transfer. Atoms, organisms, and societies are concrete systems. Humans, Jupiter moons and flowers are concrete systems. Bunge views social systems as concrete, since they are made of concrete components (i.e., marriage involves two or more physical systems) [93].

A *conceptual* or *construct-system* (see, [88]) is a system in which all of the components are mental ideas bound together in the mind of a thinking system (e.g., human being) via mental rules. Propositions, classifications, and theories are conceptual systems. Unlike concrete systems, conceptual systems do not harbor energy and change when they are changed by concrete systems.

Bunge suggested that, to represent a concrete system, four elements need to be described, namely, Composition, Environment, Structure and Mechanism of the system, which are referred to as the ***CESM model***. The *composition* of the system is its components; the *environment*, the external systems (some of which may be ill-understood or ill-defined) with which the system and its subsystems interact; and the *structure*, the relationships among its components as well as among these and the environment [111, p. 4]. Finally, *mechanism* is the "characteristic processes, that make [the system] what it is and the peculiar ways it changes" [127, p. 126]. To illustrate how to describe systems using CESM, Bunge [132, p. 39] offers among several examples, a manufacturing plant, which is a type of socio-technical system:

> *a manufacturing plant is composed of workers, engineers, and managers; its environment is a market; it is held together by contracts and relations of communication and command; and its mechanisms are those of manufacturing, trading, borrowing, and marketing.*

[3] Still, as human knowledge expands, what is considered an unrelated aggregate may later be found to be a system, a point Bunge also makes.

Conceptual systems can be represented with the condensed Composition, Environment, and Structure or the ***CES model***. Since conceptual systems do not change by themselves, they do not have mechanisms of their own. The mechanism component is not applicable to conceptual systems [129,132]. For example, a theory has components (e.g., propositions, axioms, concepts), environment (e.g., other theories that use the components of the theory in their theories, or other concepts that refer to the concepts of the theory), and structure (e.g., logical relationships linking the axioms).

Systems can also be described in terms of the *level structure* — the relationships of composition between system components [129]. Systems at one level (e.g., socio-technical, assume level 1) are composed of systems at a lower level (i.e., technical, and social, assume level 2). The social level, in turn, can be decomposed into a biological, and then chemical and psychical levels. The workers, engineers, and managers in the preceding example of a manufacturing plant illustrate the level structure as level 2 components. If necessary, we may decompose the plant example further by considering the parts of the workers, engineers, and managers such as their organs (level 3), which can be further decomposed still.

Systems have a variety of relationships with other systems beyond composition. Thus, a system can be a type of another system (e.g., bird is a type of animal, stock market is a type of social system). In this case birds share the properties of animals (e.g., multicellularity), in addition to having bird-specific properties (e.g., feathers, laying eggs).

The systems that interact with the environment are *open systems*. Likely, all systems are open, as even experimental artifacts cannot be fully isolated. However, some systems are more open and susceptible to environmental forces than others (the isolated tribes of the Andaman Islands interact with other cultures less frequently than the country of Turkey (Turkiye) which historically been the meeting grounds of different cultures).

Open systems have *boundary* — those components of the system that directly interact with the environment, whereas components that only interact with other subsystems of its parent system are *internal components*. For example, a manufacturing plant is an open system, whose boundaries include legal, HR, public relations and supply and customer service employees, among others. The internal components of a manufacturing plan include its line workers, security, and control operators, among others. These people generally do not interact with the environment as members of this system. By interacting with other systems, the systems may alter the properties, components, structure, or mechanisms of these systems. For example, when a plant produces a car, it may trigger a desire to buy the car on the part of prospective consumers.

According to Bunge, mechanism harbors the clues for why a system behaves in a particular way. To reveal the mechanism of a system is to provide an explanation for *how and why* the system works as it does, which is referred to by Bunge [96,133] and others [134] as *mechanismic explanation*. For concrete systems, the mechanismic explanations involve the description of the inner working of the system. For Bunge, this entailed detailing the different kinds of energy transfers in concrete systems, such as mechanical, thermal, kinetic, potential, electric, magnetic, gravitational, chemical (e.g., in [127]). Energy transfer leads to change in states of systems, as they acquire or lose their properties, resulting in *events* and *processes* (sequences of events). These changes may also occur as feedback loops, which, from the GST, are harbingers of natural systems.

In contrast to concrete systems, conceptual systems do not change since they, themselves, do not possess energy. However, energy transfer occurs within and between concrete systems (e.g., humans who are thinking and communicating about these conceptual systems). Conceptual systems may be externalized into some medium (e.g., paper, hard drive) in order to be communicated and shared with others, thus becoming inputs into the design of concrete systems. In this case, the *intent* to realize a conceptual systems triggers change (i.e., via energy transfer) in some concrete system, which then acts in the world to implement or instantiate the conceptual system into properties of concrete systems [135]. This can be accomplished by direct manipulation or by linguistic declarations, such as commands and requests, or *speech acts* [136–138]. For example, architectural blueprints, engineering models, and conceptual modeling diagrams, among others, originate in conceptual systems of humans. In thinking about these systems and mentally relating their properties to properties of concrete systems, humans devise means of realizing them as buildings, bridges and software code stored on a hard drive or as electrical pulses.

Note that Bunge mainly focused on changes due to energy transfer (e.g., [127]). Whether all mechanisms can be understood and modeled as energy transfer is debatable. For example, such concrete systems as governments or universities may undergo state changes driven by speech acts [114,139–142]. To describe such mechanisms, we suggest institutional ontologies that seek to understand social systems in terms of social and psychological dynamics [138,140,143]. Thus, a mechanism can be represented by physical, as well as social and psychological explanations (e.g., a contract was terminated because one of the parties *felt* dissatisfied with the terms).

Some of the energy transfers follow stable and recurring patterns, hence leading to *systemic interactions among components;* that is, those interactions that give rise to the emergent properties. For example, the working relationship among employees within an organization, such as managing and reporting functions, are systemic relationships. If removed, an organization itself may cease to function.[4]

In contrast, *ad hoc interaction among components* happen by chance, and do not follow discernible or predefined patterns. These, typically, do not give rise to the emergent properties within a system, but are still important to account for, in order to capture the full complexity of the system. For example, lending a lawnmower to a coworker is an example of such ad hoc relationship. It must be noted, however, that systems are not static, and they change, in part, when ad hoc relationships become more systemic. New systems can be born out of these ad hoc interactions.

[4] This is why some scholars define systems as assembly of components which interact with each other on a regular basis (e.g., [144,145]).

3.3. Implications of ontological systemist for conceptual modeling

Consistent with efforts to put conceptual modeling on stronger theoretical foundations, we suggest greater consideration of systems during conceptual modeling. First, systems of all kinds may need to be represented in a conceptual model using one or more constructs leading to the notion of *system as a conceptual modeling construct*. System as a conceptual modeling construct is a representation in a conceptual modeling artifact (diagram, narrative, use case) of a system as perceived by the designer or elicited from relevant stakeholders. For example, a conceptual modeling diagram may contain a system construct which is assumed to represent a manufacturing plant (a real-world system).

Second, to represent aspects of a system, one or more systemist conceptual modeling constructs may be used. These are ***systemist conceptual modeling constructs***. These constructs may present different views of the same system, as per, CESM/CEM model. For example, a conceptual modeling diagram may contain a construct (e.g., part-of association), which is assumed to represent a component of a manufacturing plant.

Third, to ensure the systemist conceptual modeling constructs cover important aspects of systems, we suggest a formalism called ***CESM+***. CESM+ adapts the CESM (CES) models of Bunge together with other populates about systems. CESM+ is a conceptual modeling template or checklist aimed to help designers describe and model essential aspects of systems of all kinds. It suggests that for a concrete system, its Composition, Environment, Structure and Mechanism should be modeled; for conceptual systems the elements to be modeled include Composition, Environment and Structure. The *plus* suggests that, in addition to modeling the above elements, the properties (hereditary and emergent) as well as other valuable or deemed relevant facts about systems (e.g., history of the system) should be considered for modeling. Among the latter, attempts to anticipate and model emergent properties should be made.

Modeling with CESM+ amounts to providing a comprehensive view of focal systems in a domain from different and converging perspectives (i.e., knowing properties of systems allows to better understand how these systems change). As we demonstrate through a case study below, this description should guide the project development toward more appropriate database, user interface and process choices. The more we understand the relevant facts about the systems of interest, the more we are able to manage their behavior, including the possibility of anticipating the elusive emergent properties.

To realize CESM+ for a given system, multiple systemist conceptual modeling constructs are needed and multiple conceptual modeling diagrams may be required. This is consistent with the growing trend toward multi-model and multi-representation conceptual modeling [146–150] and model-driven architecture [117,151].

At the same time, note that, although Bunge admits that every constituent in reality may be a system, it is important to underscore that conceptual modeling is a social activity. It models reality as perceived by human stakeholders, reflecting their needs and views of the domain. Therefore, even though every*thing* may indeed be a system from a strict materialist point of view, this does not imply that human stakeholders would conceptualize these entities as systems (mental abstractions) or automatically benefit from modeling these entities using system and its related constructs.

We suggest a more nuanced perspective, in line with, for example, [6,106]. Skyttner [106, p. 16], suggests that "[a] system is not something presented to the observer, it is something to be recognized by him/her. Usually, this word does not refer to existing things in the real world but, rather, to a way of organizing our thoughts about the same". These mental models are effectively conceptual systems glued together by mental rules. The conceptual systems may or may not accurate or completely map to properties of the concrete systems, nor even have concrete counterparts. Ptolemaic and Copernican models of the universe are example of these systems-mental abstractions. Both proved useful, despite one being less accurate than the other. Likewise, organizational stakeholders who provide information systems requirements, may have different models of systems which may be important to capture. These models of systems may not agree with one another. An open challenge is to reconcile these differences into a unified conceptual model which is effective and acceptable by the stakeholders for facilitating development and use of technology (e.g., see [152]).

Although we appreciate Bunge's basic postulate of the world of systems and maintain that a systemist approach is fruitful but underutilized in conceptual modeling, we do not suggest the system construct be immediately applicable to all modeling scenarios. Systems as modeling constructs are only useful when the systemist properties of emergence, CESM and other system-related notions are valuable to consider and, when possible, represent. Table 1 details the key systems concepts adopted and adapted in the paper within the context of conceptual modeling.

4. Illustration and further elaboration: Modeling with and without systems

Equipped with the basic ontological notions, we now discuss further the implications of these ideas for conceptual modeling. We use a case study to elaborate the representational benefits of systems and to illustrate the implications of not representing systems explicitly.

4.1. Method

We draw on a real case of information systems development within the context of online citizen science. The first author of this paper has been the primary developer of a citizen science platform, NLNature (formerly, www.nlnature.com), between 2009 and 2022. This author developed the platform, initially as a developer, hired by a biology department at a mid-sized North American university. Consequently, the author conducted the initial planning, requirements elicitation analysis, prototype development,

Table 1
Key definitions related to systems as used in this paper.

Concept and its sense, where applicable	Definition	Reference(s)	Examples
System "Ontological System" or "system out there" *an object in the world*	Complex object every part or component of which is connected with other parts of the same object in such a manner that the whole possesses some properties that its components lack — that is, emergent properties	Bunge [93, p. 20], Weber [75] and Wand and Weber [153]	Atom, animal, airplane, university, stock market, galaxy, ERP, Google
System "System-mental abstraction", "conceptual system" or "construct-system" *a potentially useful abstraction to reason about the world*	A mental model of some part of reality which *refers to* some ontological system, existing or imaginary. *Note*, conceptual systems are part of reality, being property of humans who conceptualize these systems to organize and act in the world	Skyttner [106], Luhmann [6], Bunge [111] and Lukyanenko and Weber [88]	Ptolemaic model of the universe, Copernican model of the universe, model of local biodiversity as understood by a biologist, model of Tolkien's lore as understood by a reader, theory of gravity, CESM model of a factory, the periodic table of the (chemical) elements
System "System-conceptual modeling construct" *a proposed here conceptual modeling construct*	A representation in a conceptual modeling artifact (diagram, narrative, use case) of a system as perceived by the designer or elicited from relevant stakeholders	The definition proposed in this paper; implicitly adopted in some conceptual modeling languages (see Section 2.3)	Level Structure Model (LSM) components, modeling a system using UML stereotype, proposed later in the paper
Systemist conceptual modeling constructs or systems-constructs	Conceptual modeling constructs which represent different aspects of systems, such as CESM or emergent properties	The definition proposed in this paper	*part-of* construct in UML and SysML, modeling aggregate and emergent properties, proposed later in the paper
Property and attribute	Feature, trait or characteristic possessed by a system Note: attributes are human conceptualizations of properties of systems; here, used synonymously with properties	Bunge [96,132] and Lukyanenko and Weber [88]	Mass of human, word count of a novel, color of vehicle, age of university, shape of a mathematical function
Hereditary or aggregate property	A property of a system that belongs to a component of the system	Bunge [111,132]	Income of a family member, mass of airplane components
Emergent property	A property of a system that does not belong to any of the composing parts of the system that arises when the components are bonded together	Bunge [127,129]	Cohesiveness of water, productivity of firms, consistency of theory
Aggregates or collections	To the best of existing knowledge, unrelated (i.e., not directly and continuously interacting) things and systems	Bunge [111,129]	{Jupiter's moons and Mario Bunge}, {ER 2022 conference and 3}, pile of cellphones thrown into a recycling bin
CESM and CES Models	Ontological postulate that to effectively describe a system, one needs to represent Composition, Environment, Structure and Mechanism of concrete systems and Composition, Environment and Structure of system-constructs. The *composition* of the system are its components; the *environment*, the external systems with which the system and its subsystems interact; the *structure* is the relationships among its components as well as among these and the environment, the *mechanism* is the characteristic processes, that make the system what it is and the peculiar ways it changes	Bunge [96,127]	Composition, environment, structure and mechanism of a biological family or composition, environment, and structure of a theory of thermodynamics

(*continued on next page*)

conceptual modeling, design and implementation, maintenance, and several redesign phases of the project. Later, the author joined the research team of the project. These emic experiences permit a rich insider account of modeling with and without systems. At the same time, the two remaining authors were not part of the project, hence, adding a less involved and biased perspective to the following analysis.

Table 1 (*continued*).

CESM+ Modeling Template	A conceptual modeling template used to describe essential aspects of systems of all kinds. A systems-grounded conceptual modeling should strive to represent Composition, Environment, Structure and Mechanism, properties (hereditary and emergent) as well as other relevant facts about systems	Adaptation of CESM/CES ontological models of Bunge	CESM+ is discussed and illustrated later in the paper

The NLNature project did not adopt the ontological systemism perspective, as described earlier, and used traditional (as well as experimental) systems analysis and design approaches. We, thus, engage in *retrospective analysis* — a common method in project management and systems analysis and design which draws insights from post-mortem evaluations of the successes and failures [154,155]. Specifically, we evaluate the outcomes of the project in light of the systemist ontology provided earlier and suggest implications for conceptual modeling research and practice.

4.2. Project description

The NLNature project began in 2009 with the aim to develop a citizen science platform for a region in North America. Citizen science refers to participation of the members of the general public (citizens) in scientific research, including data collection, analysis, and, more rarely, project ideation and publishing of scientific articles together with the scientists [156–159].

Since the advent of the Internet, online citizen science is emerging as a major societal movement and research approach. For example, Zooniverse.org is a citizen science platform with over 1.6 million registered users. The citizens, members of Zooniverse, work on over 50 research projects, ranging from classification of galaxies and identification of animals in the African savannah to deciphering ancient texts and locating craters and boulders of the Moon. While Zooniverse is the largest citizen science platform, it is estimated that there are over 3000 active citizen science projects. These are mainly local projects interested in specific research questions.[5] One such platform is NLNature — a representative example of a mid-sized regional citizen science platform. Indeed, the project was the regional node of a national citizen science network and the principal citizen science platform for its region of North America.

The objective of NLNature was to collect sightings of plants, animals, and other taxa in the local region (area of over 400,000 km^2). The aim was to create an evolving database of citizen-reported wildlife to support research and policy making related to plant and animal distributions, environmental change, impact of anthropogenic factors on natural habitats, and monitoring and conservation of specific species of interest, such as endangered lichens. Another goal was to raise awareness of local natural history among the residents and tourists of the region.

To support NLNature's objectives, a target list of species was identified, which became the focus of the first development stage. Upon subsequent analysis, it was clear that non-experts struggled to report their sightings using this list, so a new development philosophy was pursued whereby the citizens were permitted to report their observations without classifying the phenomena as specific species. The species could later be identified using artificial-intelligence-based techniques, such as machine learning and natural language processing. Indeed, this second phase was when one of the co-authors of the paper switched roles from developer to co-investigator to spearhead this approach to citizen science.

The project was sponsored by academia, which is typical of citizen science projects [160,161]. What makes this setting especially interesting is the nearly full transparency of the project development — part of the general commitment to open science (of course, guided by ethical protocols and appropriate participant consent) [162–165]. This permits the kinds of revelations that might be difficult to achieve when working in corporate settings.

The project has been active from 2010 to 2022. During this period, over 10,000 members registered an account on the platform. They contributed over 10,000 observations of wildlife, making nearly 3000 comments on existing observations and posting over 15,000 photos to accompany the sightings. These sightings received over 10 million user-views. Some of the observations led to scientific discoveries and resulted in several publications in scientific journals [166,167].

Over the years, NLNature underwent two major redevelopment phases with the aim to better meet the project's objectives. Consistent with prevailing approaches to citizen science development [168–170], initially the project was developed by focusing on the needs of the project sponsors: the scientists. Consequently, most of the requirement elicitation and analysis efforts concentrated on capturing the requirements of the scientists. Early-stage interviews and focus groups with the citizens were also conducted. The scientists insisted on the major unit of data collection and analysis of the project — *biological species* (e.g., Lung lichen, American robin), which became the focal entity type of the platform. Consequentially, the citizens were asked to report their observations in terms of the biological species observed. Fig. 1 shows a fragment of the conceptual model of the project showing how birds are classified per requirements of the scientists, and the resulting user interface options.

An evaluation phase began as soon as the project was launched and revealed limitations and negative consequences of approaching citizen science by privileging the views and requirements of the scientists. Specifically, non-experts could not always positively identify what was observed. Hence, while non-experts could confidently identify familiar species, such as American robins, they struggled to positively identify lichens, or unfamiliar plants and birds. The analysis of the project logs revealed that often users

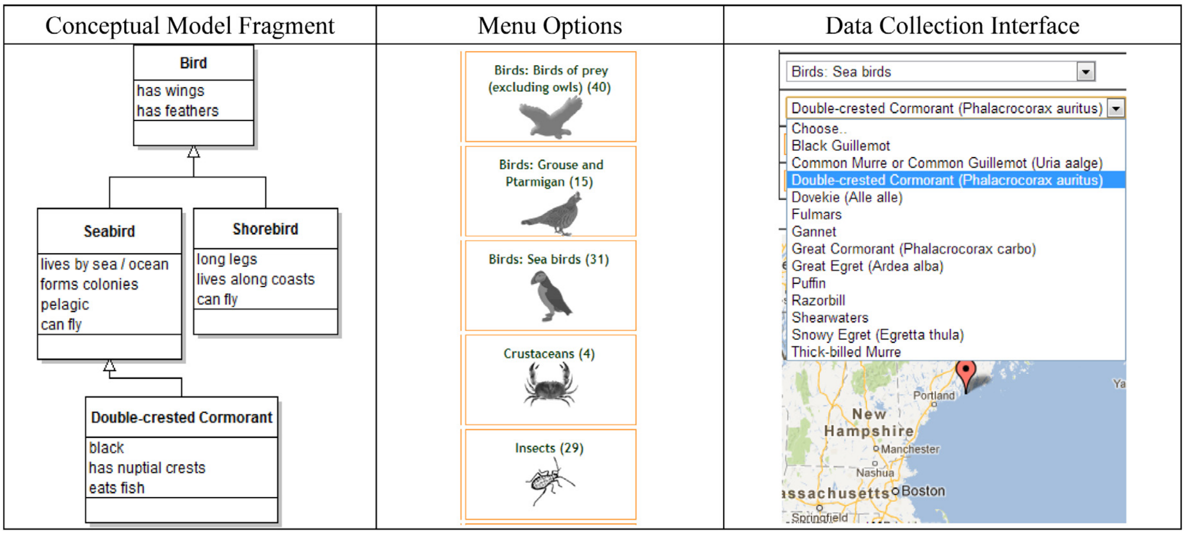

Fig. 1. Connection between a conceptual modeling fragment (showing how birds are classified per requirements of the scientists) and user interface design in Phase 1 of NLNature.

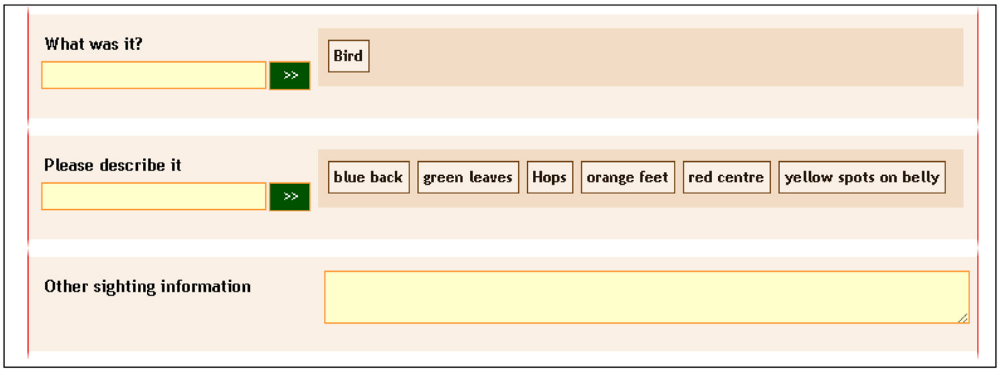

Fig. 2. User interface in Phase 2 of NLNature project with a sample observation.
Source: Taken from www.nlnature.com.

resorted to guessing, which was evidenced by frequent changes of the species field while reporting the sightings. This evidence was further supported through the analysis of user comments and interviews with the existing users [171].

In 2013 the project was redesigned, but this time consistent with an underlying ontological foundation. We chose the BWW ontology [86] as a guide for this redesign.[6] Following BWW, the new platform eschewed collecting observations using a pre-defined list of species and instead collected reports as unique instances, which citizens could describe using attributes or classes of their choice. After the data were captured, the scientists could infer species, using, for example, machine learning approaches. Fig. 2 depicts a data collection interface of NLNature developed following these ontological foundations. Fig. 3 shows a basic conceptual model of the redesigned NLNature's database to accommodate the new redevelopment philosophy.

To evaluate the utility of these ideas, we conducted a series of experiments and focus groups with the prospective and existing NLNature users. Collectively, these studies showed that data collection focusing on instances resulted in more observations being recorded, with less guessing and user frustration.

[5] The estimate is based on the projects listed on SciStarter, the world's largest database of citizen science projects; see https://scistarter.org/about.

[6] Specifically, the central tenet of BWW is that the world is made of things — substantial individuals. Things possess properties, which people can conceptualize as attributes. Things form classes when they share common properties. These ideas have been interpreted by researchers as the need to model individuals (instances) irrespective of the classes they belong to Parsons [172] and Parsons and Wand [51]. The redesign of NLNature was inspired by these ideas and interpretations.

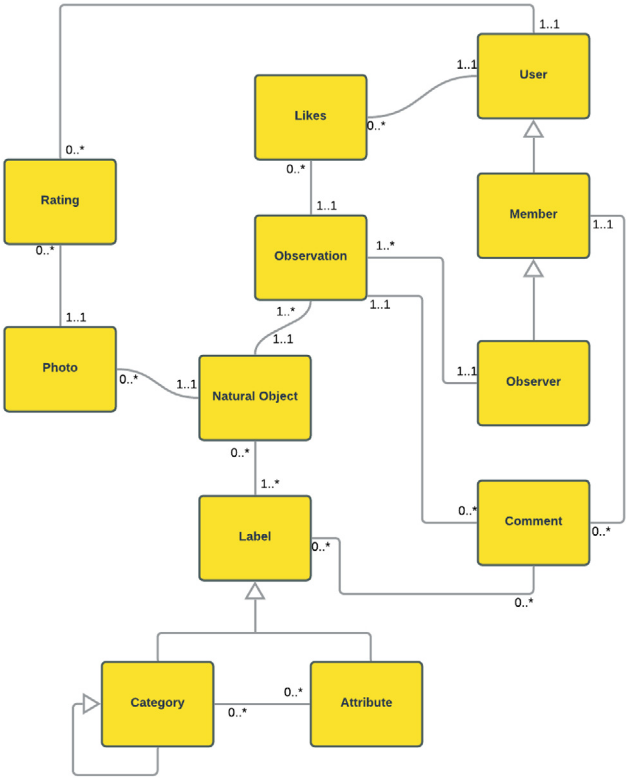

Fig. 3. A conceptual model in Phase 2 of NLNature. . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.3. Systemist analysis of the project

The systemist perspective enables us to better understand the shortcomings of modeling agnostic to systems. First, we briefly analyze the first phase of the project, before turning attention to the second one and offering a deeper systemist analysis.

4.3.1. Phase 1 systemist analysis

The benefits of modeling with systems can be evident in the simple case of Fig. 1 from the first phase of the project. Equipped with the systemist ontology (Section 3), we can now interpret this figure in systemist terms. Fig. 1 is an externalized *conceptual* system realized as a UML diagram (concrete system, captured on paper and then in software). The diagram represents concepts and hierarchies, or conceptual classification systems, of the domain of interest of NLNature, elicited mainly from the scientists. This diagram ultimately informed the development of *concrete technical sub-systems* of NLNature, a complex *socio-technical* system. These included psychical *user interface* and *database structures* (electric and magnetic charges representing binary instructions and containers) for processing and storing citizen observations of nature, developed in accordance with the concepts and hierarchies shown in Fig. 1.

Assume Fig. 1 fragment is representative of a complete model of the conceptual classification system upon which the user interface is built to collect sightings. From the systemist perspective such model shows some of the composition and structure of this conceptual system. The composition are the classes in the diagram (e.g., bird, seabird). The structure is the type of mental rules connecting these classes — that of inheritance where each class is a type of another and shares the properties of its parent.

The diagram does not show the environment. In particular, the diagram does not show the citizen scientists and others contributing or using the information organized by these classes. It neither models these systems, nor shows which part of the conceptual system (and hence, the user interface), with which they interact. In systemist terms, Fig. 1 does not show the system boundaries; that is, the classes that become the entry points for the citizen scientists (and others) into this conceptual hierarchy. The figure also does not show any emergent properties of the conceptual system, or the components of the socio-technical systems shaped by this diagram.

The lack of modeling of system boundaries and of emergent properties may complicate building accessible and usable interfaces. This realization became apparent to the project development team only after several years of NLNature's deployment. When the first phase of NLNature was launched, much of the user frustration and attrition was attributed to the lack of domain expertise on the part of less knowledgeable citizens. This, we reasoned, manifested in the inability to positively identify the observed organisms

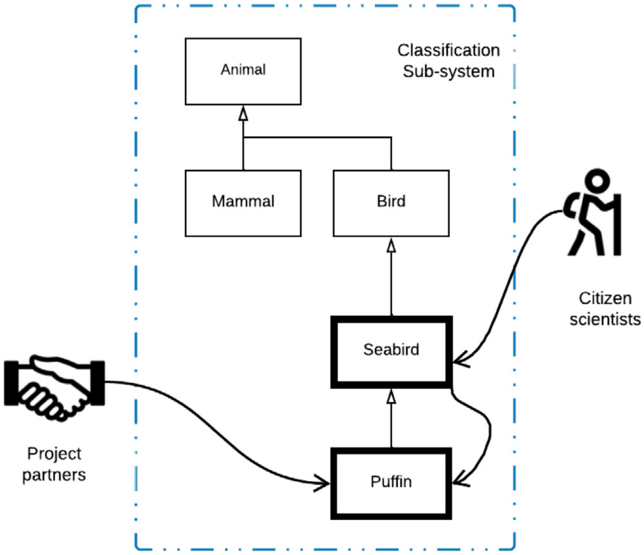

Fig. 4. Example of a System Boundary Model (SBM) fragment. Classification structure as a conceptual system (call-out bubble); boundary objects of system shown as classes with bold outlines; important environment objects shown as icons of citizen scientists and project partners.

as predefined biological species. Several laboratory and field experiments, along with user interviews, corroborated this hypothesis [167,173,174]. However, systemist perspective reveals additional causes of negative user experience, due to unexpected emergent properties and underappreciation of the systemist boundaries.

The classification system in Fig. 1 of the project contained several emergent properties. In particular, one emergent property was *overall accessibility*. In the context of the project, *overall accessibility* can be understood as the extent to which the *entire* arrangement of the classes accords with expectations and knowledge of its users, and hence is usable and accessible for reporting sightings. This is an emergent property of the sightings reporting sub-component of the NLNature socio-technical system, which was designed in accordance with the classification conceptual system of the scientists. This property emerged when all of the classes were arranged in a particular way in the user interface under the guidance of the conceptual model in Fig. 1. However, while the list of species was carefully considered (e.g., those species deemed completely inaccessible and esoteric were removed), the overall arrangement was not.

Indeed, much of research on accessible citizen science deals with a single (e.g., generic or species level of classification), not the overall arrangement of different classes [43,157,175–177]. In addition, consistent with the species focus, the impact of intermediate classes, such as Seabird or Land Mammal, was also not taken into consideration, despite these classes becoming data collection options (or, in systemist terms, system boundary).

The later analysis of the project showed that some of the negative experiences of the users were caused not by the familiarity with the individual classes, but by the overall accessibility of the classification system. For example, when reporting on a *polar bear* sighting, it was initially assumed that non-experts would be familiar with this class (which, for the most part, is a reasonable supposition). However, to get to the polar bear class, a user had to first select other top-level classes. In the project, following standard biological nomenclature [178], polar bears were modeled as a subclass of "*marine mammals*". However, for many non-experts in biology, polar bears are first and foremost, bears, and hence, land-dwellers. Hence, some users became lost upon failing to find a polar bear under the "*land mammal*" higher-level category. This and similar examples (e.g., puffins are *seabirds*, rather than *shorebirds* despite commonly being observed by the shores) reveal how the knowledge of a single system component (e.g., of a given species) may still preclude from a successful or friction-free observation due to issues in overall accessibility and failure to account for the boundary of the system. These realizations, although somewhat intuitive, escaped the original analysis.

A possible alternative to Fig. 1 diagram is shown in Fig. 4. It modifies a standard UML class diagram to depict the classification structure as a system (here: a call-out bubble); the boundary objects of the system are shown as classes with bold outlines; the main environment objects are shown as icons of citizen scientists and project partners. We call it a System Boundary Model (SBM), as its goal is to show the boundary components of systems. Boundary objects are especially valuable to model since often unanticipated events occur when components of one system begin interacting with the components of another system [179]. This gives rise to the emergent properties of the new system that forms as a result of these interactions. As a popular adage goes, innovation happens at the seams.

4.3.2. Phase 2 systemist analysis and CESM+ example

In the second iteration of NLNature, a new conceptual system was developed, shown in Fig. 3 (see earlier). The changes appeared to have addressed the shortcomings of the first design. We now apply the CESM+ modeling template to offer a general systemist analysis of NLNature as a whole.

Table 2
Analysis of NLNature based on CESM+ template.

System name and type	*NLNature, an information system, a socio-technical system*
Components	NLNature technology, composed of such components as programming code, database, application programming interfaces (APIs, such as Google Map, social media connections), and a series of hardware components (e.g., a webserver), and the social systems (scientists and citizens)
Environment	NLNature partners, media, conservation agencies, and government, as well as physical systems – objects of observation – plants, animals and other taxa observed and reported by the citizens which are analyzed by scientists
Structure	Reporting a sighting, emailing another member and asking questions, and the relationships between the users of the project and the objects in the environment
Mechanism	Making and posting of observations, reporting on information using the features of the project, communicating among project members, helping others to identify species, using the data for scientific analysis
Emergent properties (select)	• Shared sense of the project's purpose (discussed later) • Observations-anchors (i.e., existing observations which provided strong examples and influenced citizens for future observations, discussed later) • Socializing (use of platform features in unanticipated ways to elicit off-line encounters) • Research productivity • Discoveries

We suggest following a CESM+ checklist at the onset of a project. Effectively, it guides designers in considering what systems are under consideration and what components of these systems need to be considered and, possibly, represented in conceptual modeling diagrams. As with most conceptual models, CESM+ template can be applied retrospectively, to better understand an existing system (much like an entity-relationship diagram can be used to describe an existing database). Thus, Table 2 uses CESM+ template for post-hoc analysis of an existing system. By glancing at the table, the scope and essence of NLNature become more apparent.

CESM+ Analysis of NLNature. We also use the CESM+ checklist to guide the discussion on the readiness of existing conceptual modeling approaches for systems modeling. Guided by the structure and contents of CESM+ as shown in Table 2, we now: (a) offer a systemist analysis of the diagrams in Figs. 1 and 3, and (b) conduct a general assessment of existing conceptual modeling capabilities for modeling systems.

Components, Environment, Structure and Mechanism. First, without an explicit systemist perspective, many projects would lack a diagram which represents all the components of the system. Indeed, Fig. 3, which is the structure of the database, does represent some of the components of NLNature. Its vast coverage, however, may give a false impression that most focal components about which information is stored are captured. However, this is not the case. Not all informational components are shown in the conceptual model, and thus either need to be inferred, or found in other diagrams. For example, we do not see such NLNature components as the *scientists*. It should be noted that NLNature had dedicated software elements focused on scientists, such as analysis and reporting and project administration. Hence, they were objects of a database, but were modeled separately and informally. The database schema for scientists was created in an ad hoc manner without formal conceptual modeling. Not modeling some database objects is a common practice that often complicates documentation and may undermine security [180]. Had the explicit systemist perspective been adopted, such omission would be inconsistent with systemist philosophy and constitute a methodological error. *Hence, adopting a systemist perspective makes modeling more disciplined and systematic.*

Of course, it is possible to envision additional diagrams, which could represent these components. For example, an *i** Strategic Dependency Model [32,76] may include scientists together with citizens. Such a diagram may be particularly useful for understanding the goals, dependencies, and resources involved in the interaction between citizens and scientists. However, absence a systemist approach, there is no reason to expect that, for a specific project, the system is modeled as a whole and its components are analyzed and represented. The result is a lack of a holistic view of the entire system. Among other limitations, not having the holistic view of a system makes it is even more difficult to predict the emergent properties.

Absent an explicit systemist perspective (and a checklist such as CESM+), there is no guarantee that the focal objects of the environment of NLNature are modeled. These include such important entities as the federal agencies which sponsored the project, several departments within the university, and public agencies which relied on the data from the project. Among the important environment entities are other socio-technical systems that are partners of the NLNature project. For example, one such partner had an agreement to extract data related to oceans (such as ocean currents and pollution). Having this information in the model, for example, would alert systems developers as to which objects are boundary. Inattentive changes to code of such objects may undermine interoperability between NLNature and its partners. As shown in Fig. 4, it is generally possible to represent the environment, with modifications to existing conceptual modeling grammars (constructs and rules for how to use them for a particular conceptual modeling language). However, very little is known how to do so effectively, while balancing other competing objectives of conceptual models, such as parsimony and clarity.

Existing conceptual modeling languages offer support for the structure of systems. The structure is commonly modeled as relationships among system components. These can be represented using, for example, an ER diagram or UML object or class diagram. Hence, in Fig. 3, some of these relationships are shown by the associations among classes (e.g., Observer and Observation implies an *observe* relationship). Existing conceptual modeling languages can make these relationship links more explicit by naming them, as well as identifying their directions.

However, no method exists for showing the impact of these relationships on the growth and evolution of a system — a key point in describing these relationships under a systemist approach. Indeed, some of these relationships may be more important than others for ensuring system stability, whereas some relationships may be more transient and ad hoc, with less impact on the longevity of the system. Intuitively, we can reason that the posting of comments is less important than the posting of observations. Knowing this, suggests additional care and resources dedicated to the observation (technical) sub-system of the project, compared to the comments sub-system. This could be an important information for developers who lack deep domain knowledge (which can be the case in outsourcing) [181–183].

The final CESM component is mechanism. For example, the making and posting of observations is a key mechanism, which, if absent, or substantially impeded, nullifies the entire project. For the NLNature system, this is a foundational mechanism. Indeed, the evolution of this mechanism accounted for most of the code changes during the different iterations of the project. Again, such realization could help prioritize development efforts and resources. In contrast, the mechanisms involved in contacting other members are secondary to the project, and thus, are on the periphery of the NLNature system. Hence, any changes to this mechanism may occur without affecting the functioning of the entire system, and may not require as much care.

Modeling of mechanisms is possible using, for example, process oriented conceptual modeling languages, such as BPMN, EPCs, or statecharts. However, these notations are not specifically designed for representation of mechanisms, as understood by the ontological theory. Rather, they are focused on the representation of information flow or decision logic. From the point of view of systems theory, mechanisms are the explanations for *how* and *why* the system works and evolves. The process models are presently equipped at handling the *how* part (see, e.g., [184]). They do not deal with the why. For example, these models would not explain why some observations by one member were similar to observations by another, which, we hypothesized were due to anchoring effects, as discussed below. Potentially, other conceptual modeling languages, such as goal-oriented, or actor-oriented, languages and narratives [147,185], may be used for the why question. The challenge then becomes to combine the how and why in an effective manner. There is no answer to this in the extant conceptual modeling theory.

Sub-systems. Many components of the NLNature system can themselves be modeled as systems following own CESM+ template. Fig. 3 shows some of these distinguishable sub-systems, such as user observation system, user communication system, or classification system, among others. Of course, even users themselves are systems. However, it is difficult to imagine a scenario where modeling them as systems may be advantageous for this project; yet this may become important for other projects. Indeed, such crowdsourcing platforms as PatientsLikeMe (https://www.patientslikeme.com), the world's largest personalized health network that helps people find new treatments [186–189], may benefit from modeling human organs and tissues. When beneficial, CESM+ can be applied recursively, to model these subsystems. Notable here is existing conceptual modeling language do not explicitly have an ability to connect these different *CESM+* representations together.

We now illustrate modeling challenges related to one specific sub-system: user observation system and its referent objects in the NLNature environment. It was implicitly clear to the development team and the scientists that the plants, animals and other taxa represented by Natural Object in Fig. 3 are complex; that is, systems. However, they were all modeled as individuals, atomic entities, since, for a project which had hundreds of species, it was not practical to have hundreds of conceptual models of puffins, lung lichens, polar bears, and others.

Still, it could have been useful to indicate that the organisms of the project were systems, without engaging in full-blown complexity modeling. As a result of modeling entities in Fig. 3 as structureless classes, neither the database structures nor the user interface supported the collection and storage of the attributes based on the parts of the organism (system) being described. Hence, some of the attributes reported were applicable to the entire organism (e.g., large, beautiful), whereas other attributes were attributes of the parts (e.g., blue beak, yellow feet). This meant that interpreting these attributes was difficult, because it was not intrinsically clear (especially when the processing was done automatically, without human interpretation) whether this was an attribute of the entire organism or its parts.

Furthermore, frequently, the organisms reported were observed as part of larger biological systems. This too escaped the appropriate capturing by the interface which implemented the conceptual model in Fig. 3. To illustrate, Fig. 5 provides three observations. Since the project was modeled on the premise of representing individuals, it was very difficult to represent the object of these sightings as systems.

Modeling all NLNature organisms as systems was not necessary. However, had there been an ability to simply convey that the Natural Object in Fig. 3 was a system, it could have sent a signal that more complexity needed to be represented in the database. This could have been achieved by having flexible interface choices permitting, for example, key–value pairs of attribute-system parts. These could be stored following a key–value pair data model, such as that of AmazonDB or MongoDB, which permits unbounded variation, thus supporting the system diversity of NLNature [190,191]. As we can see from this analysis, adopting a systemist perspective does not always entail elaborate system modeling. Small signals from a conceptual modeling diagram, when appropriately interpreted, can be valuable. However, to create and appropriately interpret these small changes to modeling diagrams, an update to conceptual modeling methods is needed.

Emergent properties. Finally, the CESM+ checklist suggests to consider and attempt modeling emergent properties. To appreciate the benefits of such modeling, we now consider some emergent properties of NLNature (see Table 2). Two notable emergent properties, which became apparent after the implementation, are the *shared sense of the project's purpose* and *observations-anchors*. Specifically, by design, the project was intentionally broad and accepted all kinds of organisms in the specified geographic area. Over time, as citizens reported thousands of organisms, it became clear that the project began to acquire a crowd-generated identity.

We further hypothesized, this *emerged* project identity was based on certain popular observations that acted as psychological anchors [192,193]. These observations shaped the perception of what is interesting to observe, how to describe organisms, and potentially affected subsequent observations [167]. An analysis showed that most of the observations on the project were of

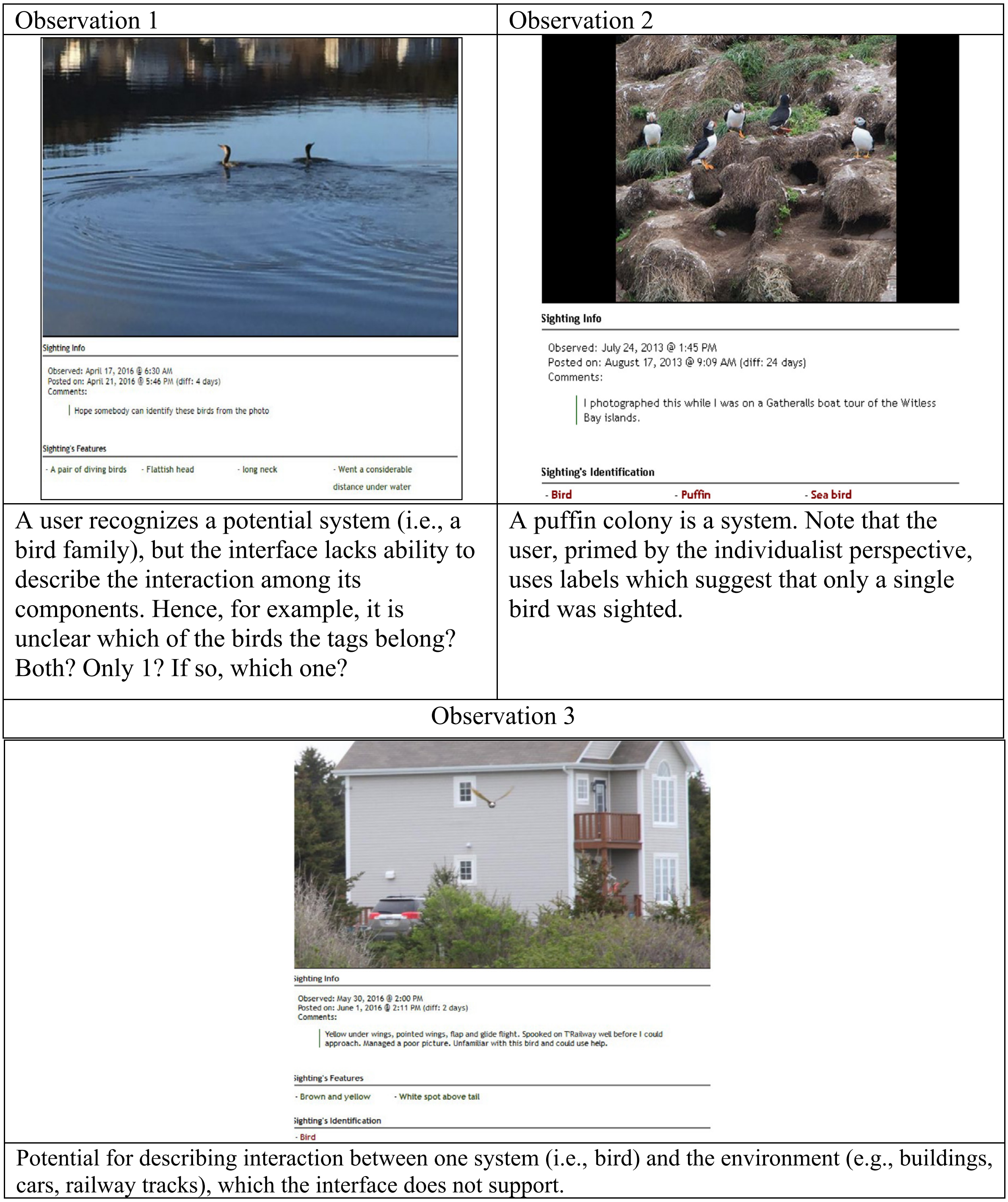

Fig. 5. Real observations where users intended to provide more descriptions of systemic aspects but could not do so.
Source: Taken from www.nlnature.com.

charismatic species, such as fox, eagle, moose, coyote, bear, mink, and seagulls. We suspected having these observations publicly visible created a grass-roots project identity and biased future observations.

This is not what the project owners wanted. They had hoped to observe a more uniform and representative map of sightings. The *shared sense of the project's purpose* emergent property was not modeled in advance, and hence no mechanisms for promptly detecting and correcting the drift toward charismatic species was envisioned during the development of the project. The paucity of

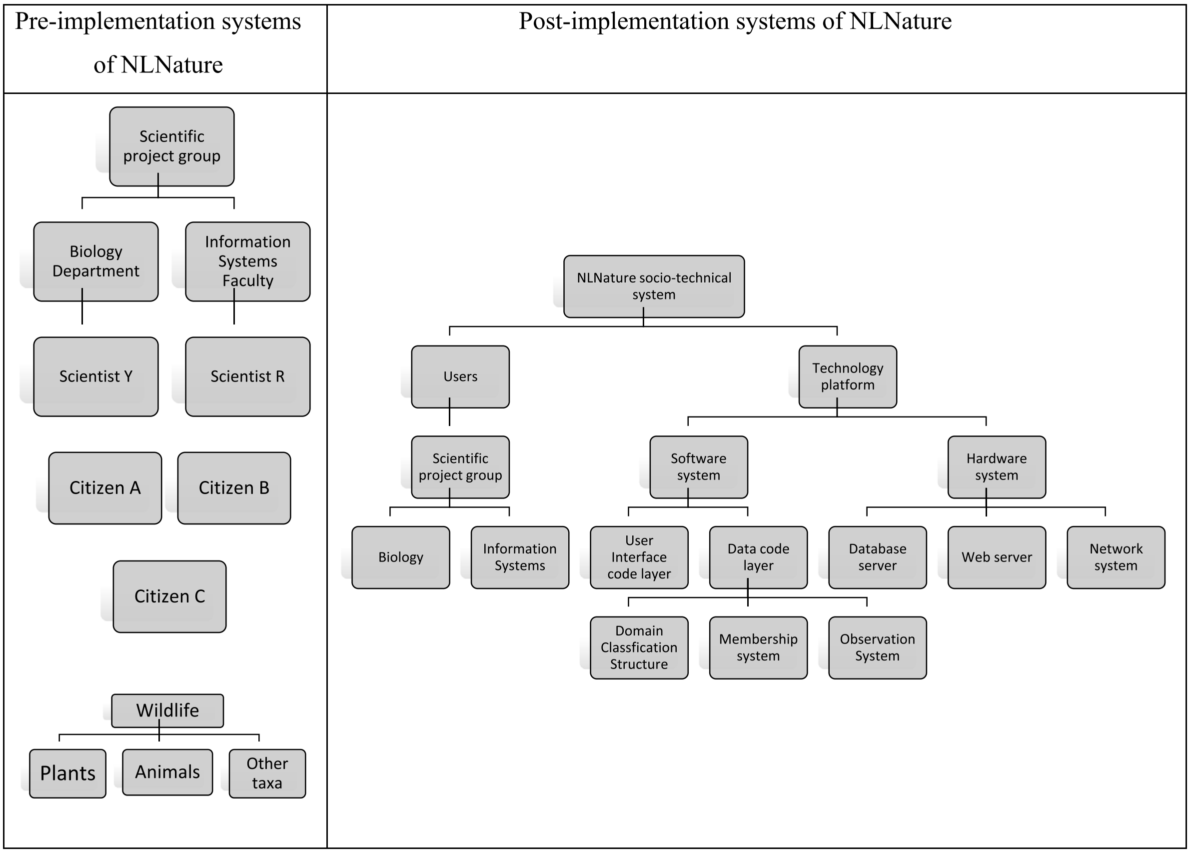

Fig. 6. Level Structure Models of pre-NLNature systems (left) and NLNature socio-technical system (right) (adapted formalism from [75]); boxes represent systems at different levels; lines represent composition relationships. Note: in NLNature user interface and data code layers were designed as separate and interacting subsystems, to permit implementation on multiple devices.

systemist thinking during conceptual modeling dissuaded the conversations about emergent properties of the entire project, as well as its subsystems.

Presently, conceptual modeling lacks established and robust abilities to detect and model emergence. In the context of the project, for example, observations-anchors can be shown as asterisks after the names of classes in a UML diagram. However, this does not permit to quantify the bias due to these anchors. A more comprehensive representation could be based on the visualization of Markov chains [194] known as *Markov network* [195]. Markov network is an undirected graphical model used to visualize stochastic processes as a sequence of possible events where the probability of an event depends on the previous event [196]. Applied to NLNature, Markov networks could model how a user viewing a set of popular observations, then has a certain probability of reporting observation with similar properties.[7] The analysis of the entire network can then shed the light on the emergence of the shared sense of project's identity. While Markov networks as a solution would not work for all scenarios, it offers a glimpse of the opportunities involved in modeling emergent properties.

Level Structure Model. To ensure complex objects are considered in modeling, it would have been helpful to have a map of components conceptualized in a project as systems. Presently, established conceptual modeling approaches do not permit such explicit expositions. To appreciate how such diagram could be constructed, we introduce a systemist diagram designed to show the components of a target system. We call it, *Level Structure Model* (LSM) of systems with the representation adapted from a formalism in [75, Chapter 2]. The LSM shows the main higher-level systems in a project. The goal of LSM is to depict the horizontal relationships between system components related via composition.

There can be multiple versions of an LSM diagram,[8] as the project progresses from the problem to the solutions space. To illustrate the usage of level structure models, Fig. 6 shows an LSM of NLNature before and after its creation. Indeed, before NLNature is

[7] We showed this effect through a controlled field experiment on NLNature [167], although it did not involve visualizing using Markov networks.

[8] The LSM envisioned here is based on structural decomposition — based on hierarchical relations among sub-systems. Other variations of component diagrams are possible, such as those based on functional decomposition — the modeling of systems based on the functions they perform [2]. In organizational design, which also deals with systems, it was found fruitful to combine structural and functional models (of firms) into matrix models [197] — a solution which may prove useful for conceptual modeling as well.

implemented, the corresponding socio-technical system does not exist. The socio-technical system arises only when technology and people begin to work together to contribute observations of wildlife, as well as to use these observations in their research activities. Hence, an early LSM version (left) shows two disparate systems — the scientific group and a single box for citizens, as although a citizen is a socio-biological system, citizens are not organized into cohesive systems. With respect to one another, they are aggregates. Once NLNature is born, citizens become linked with other systems via the technological platform. These observations enabled by the simple LSM fragment are striking because they help explain some of the future dynamics of NLNature, such as the difficulty in reaching citizens, attracting them and motivating them to join and continue contributing. Furthermore, the LSM also shows that citizens in this domain do not form a supersystem with the wildlife, which means they are not intrinsically connected with the plants and animals. Once NLNature is put in place, it calls upon the people to go out and observe their surroundings.

Finally, this simple diagram underscores the critical importance of design choices for these types of technologies. With the weak links between citizens and scientists, the technology is a key mediator between them. Any technological barriers in communication become difficult to detect and overcome. Furthermore, absent NLNature, the incentives for citizens to make observations may be removed. The second LSM model reinforces these observations.

The post-implementation LSM model offers a high-level overview of the new socio-technical system. From LSM, we can quickly ascertain the components of NLNature we choose (e.g., based on stakeholder input and domain analysis) to conceptualize post-hoc as systems in order to reveal their complexity. Hence, Fig. 6 shows that the scientists formed a social system of their own, broken down into two departments, biology and information systems. Indeed, the scientists created a tightknit network around the project, shepherding its development and evolution.

In contrast, citizens were geographically dispersed, and largely anonymous to each other (and, to a degree, to the scientists). Unlike other Internet platforms, such as social media websites Facebook, Twitter or Youtube [198–200] or collaborative crowdsourcing, such as Wikipedia [201–203], *by design*, citizens never had an organizing system of their own. Any networks and connections grew organically by finding secondary uses of the design platform features. LSM shows this by not modeling a separate citizens sub-system of NLNature. Effectively, citizens, unless self-organized, were aggregates, parts of NLNature individually.

By analyzing the LSMs in Fig. 6, we better appreciate the interaction dynamics of the project, including its information and power imbalances. The development decisions taken in the past can be better understood with this hindsight model. Indeed, for the developers, it was much more straightforward to adopt a scientist view of reality (as in Phase 1 development), when the scientists were a tightknit and well-organized group. In contrast, conducting requirements, and then reconciling and modeling goals, values, needs of the highly dispersed, heterogeneous, and unorganized citizens presented a significant challenge. With no organization of their own, the voices of citizens were *systematically* ignored. This is a notable hindsight, which matches findings on power imbalances and conflicts in online communities, open source software, and other development settings [204–208].

4.4. Case conclusions and further implications for conceptual modeling

From the analysis of the case of NLNature development, the following conclusions can be drawn, with implications for conceptual modeling. First, we conclude that just about any entity can be conceptualized as a system. Based on Bunge and modern science, systems are considered omnipresent, and can be found in almost any conceptual modeling diagram. In Fig. 3, strictly speaking, this includes all classes of the diagram. This, of course, does not mean that every class needs to be shown with a system construct. However, as the case illustrates, when the complexity of some of these objects becomes important to capture, representing these as systems becomes valuable.

Second, representing systems appears to go beyond merely showing the components. Hence, the tacit assumption that existing conceptual modeling constructs are sufficient for representing systems, may not hold. Note that popular conceptual modeling languages have used the composition construct to depict the relationship between parts and wholes [65,209,210]. Representing systems also involves capturing the environment, the structure and mechanism of a system, the system's boundary, the internal components and emergent properties, among other things. This is not often done in projects, hence, the CESM+ checklist can make modeling more disciplined and methodical.

Third, a key notion of emergence carries implications for conceptual modeling. As the ontology suggests, emergence is something that happens when the components are put together and begin to operate as a whole. Emergent properties are not directly, or easily, deducible from the properties of the components, because they emerge over time, as shown in the NLNature case. Herein lies a grand challenge: conceptual modeling occurs at the early stages of information technology development — before the IT is put to use. This means that, as information systems development shapes systems (e.g., work systems, enterprise resource systems, e-commerce systems), the a priori modeling of emergent properties may be extremely challenging, if not impossible. Hence, potentially critical properties of the systems developed with the help of conceptual modeling may escape modeling, and emerge afterwards.

Fourth, an important aspect of systems is the mechanism which, according to Bunge, gives the system its essence and is responsible for the interaction among the components as well as between the components and other systems (the environment). To capture mechanism is to explain how and why an event or process happened. For example, what is the mechanism by which social cohesion among members emerged on NLNature? Why did some observations reach identification consensus and others did not? Presently, this is an unchartered territory for conceptual modeling.

Finally, the systemic analysis does not mean that the last design iteration of NLNature was a failure. The ontological perspective taken by the project, which focused on the individuals, appeared to have addressed many of the important shortcomings of the previous approach. By focusing on the individuals, it allowed users with different backgrounds, levels of motivation, as well as familiarity and expertise with the natural history domain, to contribute observations using attributes and their own categories. Still, by ignoring systems, many valuable contributions were not appropriately captured, and many complex nuances lost. Hence, the adoption of the systemist modeling perspective, while still permitting the users to describe what they observed in terms of attributes and categories, appears to be a fruitful future design strategy.

5. Systems-aware methodological guidelines for conceptual model designers

The implicit treatment of systems ignores the fundamental ontological, and related cognitive and social status of systems in reality. While there are many outstanding questions regarding how to incorporate systems in conceptual modeling, existing conceptual modeling languages and methods already permit greater consideration of systems. By synthesizing the theoretical foundations, as well as the results from the analysis of the case study, we propose the following guidelines for conceptual model designers.

Guideline 1: Every modeling project may entail modeling systems. As the ontological theory claims, as well as evident in our case study, every entity type (or object, class), can be potentially conceptualized as a system, and hence, can be modeled using systemist constructs. Furthermore, systems may span multiple entity types or classes (discussed below), so there could be more systems that are valuable to model than there are classes or entity types. Systems are more ubiquitous than assumed by traditional conceptual modeling languages, approaches, and methodological guidelines. For some scenarios, such as those found in biology, complex engineering, or medicine, it may be prudent to assume a default status of all entities as systems. Nevertheless, *not every actual system should be conceptualized as a system*. This leads to Guideline 2.

Guideline 2: Model systems when complexity needs explicit representation. Modeling involves abstracting from irrelevant information that does not advance the purpose of modeling. The same applies to systems. Modeling something as systems should be beneficial when: the internal complexity of an entity needs to be shown; the emergent properties are important to capture; or when different system details (belonging to different levels) must be considered. In such scenarios, the additional cognitive and learning effort, as well as a potential increase in diagrammatic complexity, or the need to develop and consult additional diagrams, may be offset by the benefits of exposing the system complexity.

As we showed in the NLNature citizen science case, a useful tool for scoping the systemist analysis is the Level Structure Model (LSM) as introduced in this paper. An LSM depicts horizontal relationships between system components and provides a high-level overview of the entire system. It can be used to delineate the scope of the systemist analysis for projects, which then guides the subsequent deeper inquiries covered by the Guidelines 3–5.

Guideline 3: Follow CESM+. Once systemist conceptual modeling is adopted, analysts can follow the CESM+ checklist. Bunge's conception of systems entails describing them using the CESM model. We adapted this idea into conceptual modeling as CESM+. This new conceptual modeling formalism should act as a guide for modelers on how to approach systems-grounded conceptual modeling. It is a roadmap that can help ensure the conceptual modeling diagrams end up with a comprehensive view of focal systems in a domain from different and converging perspectives.

To realize CESM+ for a given system, multiple systemist conceptual modeling constructs are needed and multiple conceptual modeling diagrams may be required. To develop CESM+ conceptual models, analysts are advised to seek most effective and reasonable [211,212] ways to represent each element of CESM+.

Producing CESM+ conceptual models can partially be accomplished without the need to modify existing conceptual modeling grammars. Hence, the composition of the system can be shown using a *part-of* relationship in languages that support it (e.g., UML, ArchiMate). The environment may be shown as other entities that interact with the focal system (as in the example in Fig. 4).

The structure can be shown using relationships. For this, relationship-focused languages, such as ER, ORM, or UML may be used, but some extensions to these languages may be required. For example, it could be helpful to indicate whether these are *systemic* vs. *ad-hoc* relationships.

Finally, although no direct strategy appears to fully support showing mechanisms and their explanations, elements of mechanism can be shown using existing methods. For example, to show how systems conditionally react to different inputs, process models (e.g., BPMN, EPCs, petri nets) can be used. For technical systems, data flow diagrams (DFDs) could be applicable. For discrete-event systems (such as electric devices), statecharts can be applied [184,213].

To understand why systems change, languages that take a social or agent-oriented perspective are best suited. These include actor, intention and goal models (e.g., Telos, i*) [61,76,214–216]. Finally, auxiliary conceptual modeling tools, such as narratives [217,218], can also be used to capture the nuances of mechanismic explanations. Other aspects of CESM+, such as emergent properties, can be represented following the considerations provided in the next guideline.

Guideline 4: Anticipate and model emergence. The *plus* component of CESM+ suggests to model emergence. Emergent properties are not straightforward to derive and may not even manifest themselves at the time of modeling. At the same time, some strategies can be effective for anticipating and modeling emergent properties.

We suggest that designers should simulate the lifespan of a system, using tools or imagination. This can be, for example, the imagination or simulation of the implementation and usage of the artifact built with aid of a conceptual model. Here, such techniques as agent-based modeling and dynamic system simulation can be useful [99,101,219,220].

Some emergent properties can already be modeled using existing grammars. The dependencies in *i** [32,76] are indeed emergent properties that arise from the agents interacting together. Hence, at least for some domain semantics, such as those dealing with goals, tasks and resources, a Strategic Dependency models may be used.

Another potentially relevant technique is *disciplined imagination* proposed by Weick [221,222] within the context of theory development. Indeed, the anticipating of the application and use of a theory corresponds to the challenge of capturing emergent properties. In this context, the technique implies a deliberate, and persistent mental simulation of a development or use of the modeled system as an attempt to foresee its emergent behavior.

Finally, although not definitive, another approach is small-scale "pilot" or "beta" realization and deployment of the technology based on the conceptual model, in order to observe its emergent behavior *in vivo*. This technique may prove useful for some scenarios;

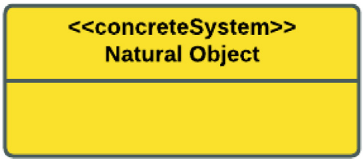

Fig. 7. Modeling a system using UML stereotype.

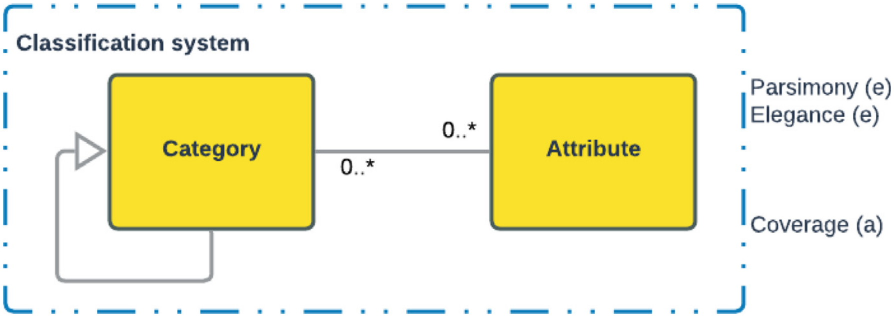

Fig. 8. Modeling a system by extending UML grammar to distinguish aggregate (a) and emergent (e) properties.

however, the behavior of a scaled-down system may not always match the behavior of the full-blown system. As Bunge explains, simulations can be valuable, but they cannot definitively capture all possible emergent properties of systems [129]. Generally, artificial systems, such as software components of socio-technical systems, under idealized and controlled conditions, are more amenable to simulation. However, the knowledge resulting from simulations of natural systems (including human social and socio-technical systems) will not offer a full view of the system since complete reduction of natural systems to its components is impossible. For a full understanding of the behavior and impact of a system, the system as a whole in its natural setting needs to be observed (see also, [223]). Still, even limited understanding of system's emergence can be much more helpful than complete ignorance.

Guideline 5: Model systems by re-interpreting or modifying existing notations

Although a comprehensive conceptual modeling support for CESM+ does not yet exist, there are, in fact, many possibilities for modeling systems by re-interpreting or making minor modifications to existing conceptual modeling languages. Below we highlight some of the possible options.

Option 1: Model using patterns or templates. Patterns and templates can be used with many existing conceptual modeling languages (e.g., UML class diagrams) to show typical, representative or anomalous systems in a domain. Hence *part-of* associations can be used to show composition; relationships can be used to show structure; activities and gateways can be used to show some aspects of the mechanism. For example, typical, or most common, species of NLNature can be modeled using patterns. For *birds*, a pattern could indicate parts, such as wings, beak, legs, and breast, which are the most common components that users describe using attributes. For *flowering plants*, stalk, leaves, and flowers could be modeled as parts. Such models could dictate the database structures and user interface features to accommodate a more nuanced user input. Hence, when a user attempts to enter an observation of a bird, a system could present a systemic schema showing the bird components and elicit attributes of the parts as well as the whole.

Option 2: A basic system construct in the diagram. In cases where is too much variations, an option is to alert the interface developers of the complexly by representing a particular class using an explicit system construct. This may or may not require significant modifications to the existing conceptual modeling grammars; that is, rules for creating conceptual models (discussed below). For example, a simple way to show a system could be based on a UML stereotype, as shown in Fig. 7. This may be sufficient in some cases as a simple way to signal complexity and potential emergence, although none are explicitly shown. Such a construct could be interpreted as, for example, the need to provide flexible database and user interface capabilities. For example, this modeling approach could indicate the need to provide key–value pairs or ontology-based data collection to better relate the part attributes to the whole; or emergent properties that are also expected as attributes of such classes.

Option 3: Extended system construct in diagram. To show emergent properties, more nuanced representations may be needed, which would go beyond merely indicating that something is a complex object. To illustrate, we propose a tentative ***multi-entity system*** construct shown in Fig. 8. The multi-entity system construct allows to represent cases when a system covers multiple entities, which in addition to being able to show system components and their relationships (or structure in CESM+), permits distinguishing between aggregate and emergent properties. Naturally, the introduction of the multi-entity system construct requires a deeper reengineering of existing conceptual modeling grammars. This is a point we revisit later.

Summary. As evident from the guidelines provided here, adopting a systemic perspective in conceptual modeling can be achieved without waiting for more extensive research on the various aspects of representing systems. Table 3 summarizes the above approaches to introducing systemist-constructs into conceptual modeling and their expected benefits. The table also outlines the general the benefits of systems-grounded conceptual modeling, based on the preceding discussion and examples. This is by no

Table 3
Examples of systems-related constructs and their expected benefits along with the benefits of systems-grounded conceptual modeling.

Diagram or pattern name	Description and possible implementation	Suggested common use cases and modeling benefits
System Boundary Model (SBM)	Representation of the boundary components of systems; These can be shown using a proposed System Boundary Model (e.g., Fig. 4) or by annotating existing structural diagrams, such as an entry-relationship or UML object or class diagram	• Boundary objects can be valuable to model since unanticipated events often occur when components of one system begin interacting with the components of another system • The potential impact of the boundaries on the ways humans interact with them can be made more explicit
Level Structure Model (LSM)	The LSM shows the main higher-level systems in a project. The goal of LSM is to depict the horizontal relationships between system components related via composition	• Provides scope of the systemist analysis (i.e., what are the systems in the domain or the system-to-be-built worthy of systemist analysis) • Offers an overview of the larger system • Can be used both before and after the system is being implemented
Systems design templates	Representations using existing conceptual modeling constructs (e.g., part-of associations) to show CESM+ components of typical, representative or anomalous systems in a domain	• When the diversity of systems is large and it is impractical to model every system, design patterns can be used for typical, representative or anomalous systems • Modeling typical or representative systems allows to signal typical use case scenarios in a domain
UML ⟪system⟫ stereotype	Shows that a class or object in question is complex, without revealing the complexity	• Alerts the interface developers of the complexly of the object, and hence, the need to have flexible input • Allows to reduce diagram complexity • Useful when diversity of systems can be large (e.g., many kinds of products), but this diversity does not need to be explicitly modeled
Multi-entity systems construct	The multi-entity system construct allows to represent cases when a system covers multiple entities	• Shows emergent properties • Shows system components and their relationships • Permits distinguishing between aggregate and emergent properties
CESM+	Roadmap and checklist for systems-grounded conceptual modeling. It is a template which reminds modelers to represent Composition, Environment, Structure and Mechanism and other facts about systems	• A guide for modelers on how to approach systems-grounded conceptual modeling • Ensures key facts about systems are considered for representation • Allows to briefly summarize key facts about systems
General benefits of systems-grounded conceptual modeling	• Greater systematization of conceptual modeling activities, especially related to representation of systems • Common concepts and vocabulary for communicating about systems of various kinds • Greater appreciation of the boundaries of systems, and the potential opportunities and challenges at the project's "seams" • Explicit representation of the often-tacit facts in a domain (e.g., systemic interactions, key mechanisms), which could help guide and prioritize development efforts • More systematic examination of emergence, with potential to anticipate potentially harmful or challenging emergent properties • More explicit understanding of the relationships between the components of systems and the emergent and aggregate properties and behavior of these systems • Better guidance for user interface and database design (e.g., by suggesting which complex objects require flexible design choices) • Increased ability to understand, create and manage social and organizational complexity	

means an exhaustive list of possible constructs to represent systems. One area of future research is to investigate these additional means of representing systems, producing an entire agenda for future conceptual modeling scholarship, which we highlight in the following section.

6. Agenda for systems-focused conceptual modeling research

Systems are the ontological primitives upon which, one could argue, other conceptual modeling constructs can be built. This, we propose, is a new approach to conceptual modeling, which brings exciting opportunities for future conceptual modeling research. Below we suggest several fruitful research directions to better incorporate systemist notions into conceptual modeling.

6.1. Research Direction 1. When to use the system construct?

Under the ontological assumption that virtually every entity in existence (even admitting a few exceptions, such as photons or quarks[9]) are systems, any object could be conceptualized by stakeholders as a system and hence may need to be represented using one or more system constructs. This applies both to modeling using abstractions (such as entity types or classes) [34,39] and to instance-based modeling, in which individual occurrences or instantiations of things form the basis for the modeling [50,51,224,225]. Yet,

[9] Subatomic particles carrying a fractional electric charge.

as stated earlier, physics and philosophy notwithstanding, "[a] system is not something presented to the observer, it is something to be recognized by him/her" [106, p. 16]. Indeed, the NLNature case demonstrated that. Over the course of ten years, multiple systems could be identified in the project (see LSM in Fig. 6). Still many more classes in the Phase 2 diagram (Fig. 3), for example, do not appear to benefit from the exposure of hidden complexities (e.g., Rating, Like, Attribute, Comment). Indeed, such unpacking of the CESM+ elements would add much overhead and complexity to the conceptual model for little evident gain.

It is likely that not every entity or object could benefit from being represented as a system. The very idea of conceptual modeling is to deliberately abstract from unnecessary, irrelevant details, and to focus only on those aspects of the domain that are important to represent for a particular purpose [8,34,226–228]. In many modeling applications, it is sufficient to represent an entity as an atomic, singular unit, rather than a complex object. It is then necessary to understand the design principles underlying the distinction between modeling parts of reality as singular entities versus modeling them as complex objects, systems.

In this paper we provided suggestions for when explicit modeling of the system may be beneficial: in cases where the consideration and representation of CESM+ components is warranted. Still, this does not exhaust the issue. Emergent properties are difficult to anticipate in advance. How would an analyst know that, for example, the innocent looking Like class, does not harbor important and consequential emergent properties? More generally, how does the analyst know that what stakeholders describe as system-abstractions (see Table 1) are indeed worthy of modeling using a system construct? By considering these questions, future research can offer a more formalized set of procedures for determining the need for system modeling and systems-driven requirements elicitation, contingent upon the specific parameters and constraints.

Once the principles for how to identify the scope of systemist analysis are established, they can inform the rules for developing the Level Structure Model (LSM diagram) introduced in this paper. Equipped with these rules, this diagram can then be interpreted with less ambiguity, as definitively representing the scope of the systemist analysis.

It may be true that there are no simple, structureless entities, and even elementary particles are complex objects/systems (i.e., composed of other systems). Therefore, this possibility implies an infinite recursion. For the majority of applications, it is not a practical challenge because it is not necessary to model elementary particles such as quarks, and then seek to model its components and then the components of these components. Yet, for those cases where modeling such entities could be needed (e.g., [229,230]), some representations of the system construct may be inappropriate. Hence, further work is needed that focuses on the problem of recursion and ways to address this issue without introducing infinite loop possibilities into conceptual modeling grammars. Such work may benefit from a long-standing debate in philosophy on the nature of infinite regress [231–235].

6.2. Research Direction 2. Development of the representation of systems

Representing a system involves more than simply identifying the component parts, as currently supported by the popular conceptual modeling languages, such as UML. Systems indeed appear to require a dedicated representation. For example, Bunge proposed the CESM model, which is also incorporated in the BSO ontology. We suggested this ontological idea could become a design template for representing systems in a conceptual model, termed CESM+. While CESM+ can immediately be a useful checklist for considering different aspects of systems (as we showed in Table 2), finding the most effective ways of representing the different components of CESM+ would require additional design work in conceptual modeling to determine how to incorporate the CESM+ components into conceptual modeling diagrams.

A pressing question is how to represent the individual elements of CESM+. Many existing conceptual modeling grammars (e.g., UML Class Diagrams) contain provisions for representing, for example, system components (via part-of relationship). However, a more challenging issue is how to represent the environment (by showing what a system is and what it is not, which requires an explicit focal system/other systems distinction among constructs). Virtually all systems are open (meaning that boundary components may interact with the environment directly), so it may be helpful to depict this explicitly.

The challenge further becomes how to represent the structure that captures the dependencies among the components. Here, for example, an objective may be to distinguish between systemic interactions (e.g., work or payment relationships between employers and employees), versus ad hoc interactions that occur as part of the system, but do not define its structure (e.g., a one-time invitation from an employee to assist with a house move). Although the *i** was developed to support modeling of actor goals and intentions [32,76], an intriguing possibility is to use this framework to capture the dependencies among systemic components.

Another challenge is how to incorporate the mechanism, which is an aspect of the system that provides its essence and carries an explanation for why the system behaves in a particular way. This provides a new avenue in conceptual modeling thinking, which expands the objective of conceptual modeling from that of representation, to also include an explanation.

Assuming additional provisions to represent CESM+, research is needed to consider the increased complexity of the diagrams so that the introduction of additional elements is clearly identified and effective visual representation of the elements is found. Such research would benefit from guidance on: managing complexity [63,236,237]; the physics of conceptual modeling notations [238]; cognitive mechanics in diagram processing [149,239]; and the evaluation of different conceptual modeling design choices [240–242].

Although CESM+ can be a series of textual descriptions describing various system components (see Table 2), based on multimedia learning theory [243,244], we can predict additional benefits from the CESM+ template if it could be depicted graphically. This could be a kind of ***Systems Canvass***, an idea akin to Business Model Canvass by Osterwalder and Pigneur [245–247]. Future research could elaborate on the idea of a Systems Canvass as a graphically-organized high-level description of a system following the CESM+ template.

The extensions of CESM+ can be investigated. For example, CESM+ does not consider the functions systems perform. Since many systems designed with the support of conceptual models are functional artifacts [66,248,249], future studies could extend CESM+ to take into account the functionality of these systems and, possibly, relate it to the other elements of CESM+ (e.g., mechanism).

We followed Bunge and suggested CESM+ as a guide for describing systems. The CESM+ model is general and can be used to model natural and artificial systems. However, other systemist models can be more applicable, especially for specific kinds of systems. An opportunity for research is investigating different approaches to representing systems, beyond the CESM model. For example, one such model is Checkland's CATWOE (customer, actor, transformation, world view, owner, environment) [97,250]. This model can be an effective representational template especially for purposeful systems that have defined owners, customers, and a world view, which is something that CESM does not distinguish (for analysis of CATWOE, see, e.g., [251,252]). Future studies can evaluate the strengths of different systemist modeling templates, which would be akin to comparisons between different general ontologies or modeling formalisms. These comparison studies are well-established in conceptual modeling research (e.g., [35,38,63,64,84,253–256]) and have developed methodological bases [240–242,257–262], which could be applied to the systemist model comparisons.

6.3. Research Direction 3. Modeling of emergence

A key notion of the systemist approach to modeling is that of emergence, which is captured as the *plus* in CESM+ modeling template. As argued and shown in the case of the development of a real information technology, emergence is an omnipresent phenomenon when dealing with complexity of real-world domains. The problem, however, is that conceptual modeling happens typically at the early stages of the information systems development and assumes a static representation of the domain.

The prevailing approaches to conceptual modeling appear ill-equipped to capture the emergence and provide the requisite support for the technology development. As already noted, in certain cases, emergent properties can be evident, especially for large complex systems, such as the entire NLNature. However, how do we know which object of modeling is void of consequential emergence? A major future research direction is how to simulate emergence, and incorporate it into conceptual modeling diagrams, methods, and techniques. There is an exciting opportunity for conceptual modeling research to collaborate with disciplines that have dealt with dynamic systems, such chemistry, physics, engineering, medicine, and social science. Solutions may be potentially found in such techniques as agent-based modeling and dynamic system simulation [219,220].

We already discussed Markov networks for some emergent properties within the context of NLNature. Markov networks have been popular in the artificial intelligence community for visualizing and modeling complex stochastic processes [195,263]. Leveraging Markov networks in conceptual modeling (e.g., as graphs supported by data produced by model simulations) may create synergies between artificial intelligence (machine learning) and conceptual modeling, which is an emerging conceptual modeling frontier [264–267].

The application of such emergence-inspired notions as disciplined imagination [221,222] can also be investigated. Finally, such frameworks as dependencies in *i** [32,76] may also be effective means of modeling certain emergent properties.

6.4. Research Direction 4. Analysis of existing conceptual modeling constructs based on ontological systemism

Existing conceptual modeling constructs can be subjected to ontological analysis, as in prior research on conceptual modeling languages [65,268–274]. Indeed, both the entity and attribute constructs may suffer from construct overload when systems are taken into consideration. The entity construct often represents atomic, as well as complex, objects. Likewise, an attribute construct may denote intrinsic, aggregate, or emergent properties. Future research can consider the implications of these cases of construct overload for domain understanding, model expressiveness, and other dependent variables of interest.

This ontological analysis could be extended to specific applications for evaluation of the effectiveness within a domain. Such research could provide an ontological explanation for existing constructs, such as dependencies in *i**, pools in BPMN, because both constructs implicitly represent aspects of systems. The new approach to explicit modeling of systems proposed in this research, can serve as a basis for further refinement of these constructs.

6.5. Research Direction 5. Expansion of existing conceptual modeling languages

Accepting the merit of using a dedicated system construct implies that existing conceptual modeling languages can be enriched through the addition of a dedicated system symbol. For example, the entity-relationship diagram could now include another major construct (system), making it a diagram that represents entities, attributes, relationships, and systems. A system can be represented as a dashed box surrounding the entity types, which are deemed as components or parts of the system (e.g., as done in Fig. 8).

The addition of the system construct leads to the rethinking of the ontological status of the entity. Once system is added to the entity-relationship model, the "entity" construct can be understood as an atomic, simple object. Everything can be deemed to be a system. However, in practical cases, where showing system complexity is irrelevant for the task at hand, we can model systems as entities; that is, structureless systems (an oxymoron, of course). For these scenarios, the construct of an entity can be uniquely reserved, and contrasted with the construct of a system, which is a construct exclusively reserved for entities conceptualized as complex. Future research is needed so systems can be incorporated into the grammars. As suggested with our analysis of the entity-relationship diagrams, this may require rethinking the definitions of existing constructs within these languages.

6.6. Research Direction 6. A systems perspective for model-driven development and MDA-based approaches

The use of a system notion can be a novel solution to the lack of a conceptual integration for the different systemic components of a real world MDD/MDA (Model-Driven Development/Model-Driven Architecture) specification [275–277]. In MDA terms, different models are at different levels of abstraction, such as Computation-Independent Model (CIM), Platform Independent Model (PIM), or Platform Specific Model (PSM). These focus on different relevant conceptual granularities, each covering a specific system dimension, whose integration is not a simple task. This is not as evident as it should be when we consider the system as a holistic conceptual unit. For instance, an i* organizational model (CIM level) [278,279] represents a goal model, whose task dependencies between actors should be described in detail using BPMN model components (PIM level dealing with system functionality). The data participating in those BPMN processes must be identified and represented in a data model (e.g., an ER model, conforming to a PIM level from the data structure point of view).

These different levels are really representing different perspective of the whole. Preserving this unified systemism perspective is crucial. This is because a conceptually grounded, sound traceability between the different levels of abstraction used in the process of describing the system is essential to achieve a sound IT design. The notion of system can help to conceptually deal with MDA-based model transformation processes, and assess their quality by providing a holistic perspective, which is too frequently omitted. Further research could explicitly consider the benefits and limitations of adopting a systemist perspective in MDD/MDA contexts.

7. Conclusion

In response to the increasing demands on IT development, this paper has argued for the need to model an overlooked notion of a system as a basic conceptual modeling construct. The system construct is firmly based on ontological principles that serve as its fundamental justification. The proposed systemist approach was illustrated through application to a case study for developing a citizen science application. Doing so has shown that modeling with greater, and explicit, consideration of systems appears to be a fruitful way to deal with our ever-changing, and increasingly complex, reality. Recommendations for future research are based on a set of specific modeling needs, namely, the need to model the complexities of the physical and digital realities. Overall, the systemist approach will require revisiting well-known and well-accepted modeling constructs to progress conceptual modeling of contemporary and future applications and, in doing so, provide new opportunities for conceptual modeling research and practice.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We wish to thank the editor-in-chief, Carson Woo, and three anonymous reviewers for their exceptionally insightful and developmental comments. The substantial improvements that resulted from their feedback were much deeper than we usually experience in journal review processes. We wish to thank the participants of www.nlnature.com (now inactive) who contributed their sightings from 2010 to 2022. We also thank Jeffrey Parsons and Yolanda Wiersma — the co-investigators of NLNature. We are grateful to the late Mario Bunge and to Ron Weber with whom we discussed ontological ideas that inspired this paper. We also want to thank the participants and reviewers of AIS SIGSAND and ER Conference for their comments and feedback on earlier versions of this paper. This research was supported by McIntire School of Commerce, University of Virginia, J. Mack Robinson College of Business, Georgia State University, United States, and by VRAIN Research Institute of the Universitat Politècnica de Valencia and the Generalitat Valenciana, Spain under the CoMoDiD project (CIPROM/2021/023).

References

[1] J. Clark, M. Cohen, The Century of Complexity, Beshara Magazine, 2017, https://besharamagazine.org/science-technology/dr-vasileios-basios-the-century-of-complexity/.
[2] J.L.G. Dietz, Enterprise Ontology: Theory and Methodology, Springer, Berlin/ Heidelberg, 2006, https://books.google.com/books?id=jg_CXuVjJyIC.
[3] A. Fayoumi, R. Williams, An integrated socio-technical enterprise modelling: A scenario of healthcare system analysis and design, J. Ind. Inf. Integr. (23) (2021) 100221, Elsevier.
[4] Y.N. Harari, Homo Deus: A Brief History of Tomorrow, Random House, London, UK, 2016.
[5] S. Johnson, Emergence: The Connected Lives of Ants, Brains, Cities, and Software, Simon and Schuster, New York NY, 2002.
[6] N. Luhmann, Social Systems, Stanford University Press, Palo Alto, California, USA, 1995, https://books.google.ca/books?id=zVZQW4gxXk4C.
[7] A. Burton-Jones, J. Recker, M. Indulska, P. Green, R. Weber, Assessing representation theory with a framework for pursuing success and failure, MIS Quart. 41 (4) (2017) 1307–1333.
[8] H.C. Mayr, B. Thalheim, The Triptych of Conceptual Modeling, Software and Systems Modeling, Springer, 2020, pp. 1–18.
[9] V.C. Storey, J.C. Trujillo, S.W. Liddle, Research on conceptual modeling: Themes, topics, and introduction to the special issue, Data Knowl. Eng. (98) (2015) 1–7.
[10] Y. Wand, R. Weber, Research commentary: Information systems and conceptual modeling — A research agenda, Inf. Syst. Res. 13 (4) (2002) 363–376.
[11] C.E.H. Chua, M. Indulska, R. Lukyanenko, W. Maass, V.C. Storey, Data management, MISQ Quart. Online (2022) 1–10.
[12] T.J. Teorey, D. Yang, J.P. Fry, A logical design methodology for relational databases using the extended entity-relationship model, ACM Comput. Surv. 18 (2) (1986) 197–222.
[13] B. Thalheim, Entity-Relationship Modeling : Foundations of Database Technology, Springer, Berlin, New York, 2000.

[14] B. Curtis, M.I. Kellner, J. Over, Process modeling, Commun. ACM 35 (9) (1992) 75–90.
[15] M. Dumas, M. La Rosa, J. Mendling, H.A. Reijers, Fundamentals of Business Process Management, Vol. 1, Springer, Berlin/ Heidelberg, 2013.
[16] J. Recker, Continued use of process modeling grammars: The impact of individual difference factors, Eur. J. Inf. Syst. 19 (1) (2010) 76–92.
[17] V. Khatri, I. Vessey, V. Ramesh, P. Clay, S.-J. Park, Understanding conceptual schemas: Exploring the role of application and IS domain knowledge, Inf. Syst. Res. 17 (1) (2006) 81–99.
[18] J. Mylopoulos, Information modeling in the time of the revolution, Inf. Syst. 23 (3–4) (1998) 127–155.
[19] K. Siau, Informational and computational equivalence in comparing information modeling methods, J. Database Manage. (JDM) 15 (1) (2004) 73–86.
[20] C. Woo, The role of conceptual modeling in managing and changing the business, in: International Conference on Conceptual Modeling, Springer, 2011, pp. 1–12.
[21] J. Recker, R. Lukyanenko, M.A. Sabegh, B.M. Samuel, A. Castellanos, From representation to mediation: A new agenda for conceptual modeling research in a digital world, MIS Quart. 45 (1) (2021) 269–300.
[22] F. Ahmed, N. Zviedrite, A. Uzicanin, Effectiveness of workplace social distancing measures in reducing influenza transmission: A systematic review, BMC Public Health 18 (1) (2018) 1–13, BioMed Central.
[23] P. Caley, D.J. Philp, K. McCracken, Quantifying social distancing arising from pandemic influenza, J. R. Soc. Interface 5 (23) (2008) 631–639, The Royal Society London.
[24] L. Matrajt, T. Leung, Evaluating the effectiveness of social distancing interventions to delay or flatten the epidemic curve of coronavirus disease, Emerg. Infect. Diseases 26 (8) (2020) 1740, Centers for Disease Control and Prevention.
[25] V.C. Storey, R. Lukyanenko, C. Grange, Fighting pandemics with physical distancing management technologies, J. Database Manage. 33 (1) (2022) 1–14.
[26] G. Tabourdeau, C. Grange, From user acceptance to social acceptance, in: SIGHCI 2020 Proceedings, 2020, pp. 1–10.
[27] P. Chen, The entity-relationship model — toward a unified view of data, ACM Trans. Database Syst. 1 (1) (1976) 9–36.
[28] G. Gottlob, M. Schrefl, B. Rock, Extending object-oriented systems with roles, ACM Trans. Inf. Syst. 14 (3) (1996) 268–296.
[29] T. Halpin, Fact-oriented modeling: Past, present and future, Conceptual Model. Inf. Syst. Eng. 342 (2007) 19–38.
[30] L. Limonad, Y. Wand, Extending business process models with controls, in: SIGSAND 2008, 2008, p. 131.
[31] C. Sapia, M. Blaschka, G. Höfling, B. Dinter, Extending the E/R Model for the Multidimensional Paradigm, Springer, 1998, pp. 105–116.
[32] E. Yu, Agent-oriented modelling: Software versus the world, in: Agent-Oriented Software Engineering II, Springer, 2002, pp. 206–225.
[33] R.C. Goldstein, V.C. Storey, Unravelling is-a structures, Inf. Syst. Res. 3 (2) (1992) 99–126, INFORMS.
[34] J.M. Smith, D.C.P. Smith, Database abstractions: Aggregation and generalization, ACM Trans. Database Syst. 2 (2) (1977) 105–133.
[35] M.I. Aguirre-Urreta, G.M. Marakas, Comparing conceptual modeling techniques: A critical review of the EER vs. OO empirical literature, ACM SIGMIS Database 39 (2) (2008) 9–32.
[36] I. Davies, P. Green, M. Rosemann, M. Indulska, S. Gallo, How do practitioners use conceptual modeling in practice? Data Knowl. Eng. 58 (3) (2006) 358–380.
[37] B. Dobing, J. Parsons, How UML is used, Commun. ACM 49 (5) (2006) 109–113.
[38] T. Halpin, Comparing data modeling formalisms, Commun. ACM 38 (10) (1995) 18.
[39] J. Peckham, F. Maryanski, Semantic data models, ACM Comput. Surv. 20 (3) (1988) 153–189.
[40] I.-Y. Song, M. Evans, E. Park, A comparative analysis of entity-relationship diagrams, J. Comput. Softw. Eng. 3 (4) (1995) 427–459.
[41] J. Parsons, Y. Wand, Choosing classes in conceptual modeling, Commun. ACM 40 (6) (1997) 63–69.
[42] V.C. Storey, Relational database design based on the entity-relationship model, Data Knowl. Eng. 7 (1) (1991a) 47–83, Elsevier.
[43] A. Castellanos, M. Tremblay, R. Lukyanenko, B. Samuel, Basic classes in conceptual modeling: Theory and practical guidelines, J. Assoc. Inf. Syst. 21 (4) (2020) 1001–1044.
[44] J. Evermann, Y. Wand, Towards ontologically based semantics for UML constructs, in: Conceptual Modeling—ER 2001, 2001, pp. 354–367.
[45] R. Lukyanenko, B.M. Samuel, Are all classes created equal? Increasing precision of conceptual modeling grammars, ACM Trans. Manage. Inf. Syst. (TMIS) 40 (2) (2017) 1–25.
[46] J. Parsons, Y. Wand, Extending classification principles from information modeling to other disciplines, J. Assoc. Inf. Syst. 14 (5) (2012) 2.
[47] O. Eriksson, P. Johannesson, M. Bergholtz, The case for classes and instances-a response to representing instances: The case for reengineering conceptual modelling grammars, Eur. J. Inf. Syst. 28 (6) (2019) 681–693.
[48] O. Eriksson, P.J. Agerfalk, Rethinking the meaning of identifiers in information infrastructures, J. Assoc. Inf. Syst. 11 (8) (2010) 433–454.
[49] R.C. Goldstein, V.C. Storey, Materialization [database design], IEEE Trans. Knowl. Data Eng. 6 (5) (1994) 835–842, IEEE.
[50] R. Lukyanenko, J. Parsons, B.M. Samuel, Representing instances: The case for reengineering conceptual modeling grammars, Eur. J. Inf. Syst. 28 (1) (2019) 68–90, http://dx.doi.org/10.1080/0960085X.2018.1488567.
[51] J. Parsons, Y. Wand, Emancipating instances from the tyranny of classes in information modeling, ACM Trans. Database Syst. 25 (2) (2000) 228–268.
[52] S. de Cesare, B. Henderson-Sellers, C. Partridge, M. Lycett, Improving Model Quality Through Foundational Ontologies: Two Contrasting Approaches to the Representation of Roles, Springer, 2015, pp. 304–314.
[53] S. de Cesare, C. Partridge, BORO as a foundation to enterprise ontology, J. Inf. Syst. 30 (2) (2016) 83–112, American Accounting Association.
[54] G. Guizzardi, G. Wagner, J.P.A. Almeida, R.S. Guizzardi, Towards ontological foundations for conceptual modeling: The unified foundational ontology (UFO) story, Appl. Ontol. 10 (3–4) (2015) 259–271.
[55] A. Burton-Jones, R. Weber, Building conceptual modeling on the foundation of ontology, in: Computing Handbook: Information Systems and Information Technology, CRC Press, Boca Raton, FL, United States, 2014, 15.1-15.24.
[56] J. Recker, M. Rosemann, J. Krogstie, Ontology-versus pattern-based evaluation of process modeling languages: A comparison, Commun. Assoc. Inf. Syst. 20 (1) (2007) 48.
[57] P. Soffer, M. Kaner, Y. Wand, Assigning ontology-based semantics to process models: The case of Petri nets, Adv. Inf. Syst. Eng. 1 (2008) 6–31.
[58] T. Wahl, G. Sindre, An analytical evaluation of BPMN using a semiotic quality framework, Adv. Top. Database Res. (5) (2006) 94–105.
[59] I. Bider, P. Johannesson, P. Soffer, Y. Wand, On the Notion of Soft-Goals in Business Process Modeling, Business Process Management Journal, Emerald Group Publishing Limited, 2005.
[60] K. Monu, C. Woo, Intelligent agents as a modeling paradigm, in: ICIS 2005 Proceedings, Las Vegas, Nevada, 2005, pp. 167–179.
[61] J. Yan, D. Hu, S.S. Liao, H. Wang, Mining agents' goals in agent-oriented business processes, ACM Trans. Manage. Inf. Syst. (TMIS) 5 (4) (2015) 20.
[62] F. Bodart, A. Patel, M. Sim, R. Weber, Should optional properties be used in conceptual modelling? A theory and three empirical tests, Inf. Syst. Res. 12 (4) (2001) 384–405.
[63] A. Gemino, Y. Wand, Complexity and clarity in conceptual modeling: Comparison of mandatory and optional properties, Data Knowl. Eng. 55 (3) (2005) 301–326.
[64] G. Shanks, E. Tansley, J. Nuredini, D. Tobin, R. Weber, Representing part-whole relations in conceptual modeling: An empirical evaluation, MIS Quart. 32 (3) (2008) 553–573.
[65] Y. Wand, V.C. Storey, R. Weber, An ontological analysis of the relationship construct in conceptual modeling, ACM Trans. Database Syst. 24 (4) (1999) 494–528.
[66] S. Chatterjee, S. Sarker, M.J. Lee, X. Xiao, A. Elbanna, A possible conceptualization of the information systems (IS) artifact: A general systems theory perspective 1, Inf. Syst. J. 31 (4) (2021) 550–578, Wiley Online Library.

[67] K. Lyytinen, M. Newman, Explaining information systems change: A punctuated socio-technical change model, Eur. J. Inf. Syst. 17 (6) (2008) 589–613.
[68] S. Winter, N. Berente, J. Howison, B. Butler, Beyond the organizational 'container': Conceptualizing 21st century sociotechnical work, Inf. Organ. 24 (4) (2014) 250–269.
[69] R.P. Bostrom, J.S. Heinen, Mis problems and failures: A socio-technical perspective. Part I: The causes, MIS Quart. 1 (3) (1977) 17–32, JSTOR.
[70] E. Mumford, The story of socio-technical design: Reflections on its successes, failures and potential, Inf. Syst. J. 16 (4) (2006) 317–342, Wiley Online Library.
[71] W.J. Orlikowski, S.R. Barley, Technology and institutions: What can research on information technology and research on organizations learn from each other? MIS Quart. 14 (3) (2001) 145–165, JSTOR.
[72] M.M. Lankhorst, H.A. Proper, H. Jonkers, The anatomy of the archimate language, Int. J. Inf. Syst. Model. Des. (IJISMD) 1 (1) (2010) 1–32.
[73] C. Atkinson, R. Gerbig, M. Fritzsche, A multi-level approach to modeling language extension in the enterprise systems domain, Inf. Syst. (54) (2015) 289–307, Elsevier.
[74] C.L. Azevedo, M.-E. Iacob, J.P.A. Almeida, M. van Sinderen, L.F. Pires, G. Guizzardi, Modeling resources and capabilities in enterprise architecture: A well-founded ontology-based proposal for ArchiMate, Inf. Syst. (54) (2015) 235–262.
[75] R. Weber, Ontological Foundations of Information Systems, Coopers & Lybrand, Australia, 1997.
[76] E.S. Yu, Social Modeling and I, Conceptual Modeling: Foundations and Applications, Springer, 2009, pp. 99–121.
[77] S. Alter, Work system theory: Overview of core concepts, extensions, and challenges for the future, J. Assoc. Inf. Syst. 14 (2) (2013) 72.
[78] S. Alter, A workaround design system for anticipating, designing, and/or preventing workarounds, in: Enterprise, Business-Process and Information Systems Modeling, Springer, 2015, pp. 489–498.
[79] S. Alter, Framework for describing a theoretical perspective: Application to the Bunge-Wand-Weber ontology and general systems theory, in: Australasian Conference on Information System, Sydney, Australia, 2021, pp. 1–11.
[80] A. Burton-Jones, C. Grange, From use to effective use: A representation theory perspective, Inf. Syst. Res. 24 (3) (2012) 632–658.
[81] C. Gonzalez-Perez, How ontologies can help in software engineering, in: International Summer School on Generative and Transformational Techniques in Software Engineering, Springer, 2015, pp. 26–44.
[82] N. Guarino, Formal Ontology and Information Systems, Vol. 98, 1998, pp. 81–97.
[83] N. Guarino, G. Guizzardi, J. Mylopoulos, On the philosophical foundations of conceptual models, Inf. Model. Knowl. Bases 31 (321) (2020) 1.
[84] G. Guizzardi, Ontological Foundations for Structural Conceptual Models, Telematics Instituut Fundamental Research Series, Enschede, The Netherlands, 2005.
[85] S.T. March, G.N. Allen, Toward a social ontology for conceptual modeling, Commun. AIS 34 (1) (2014) 1–30.
[86] Y. Wand, R. Weber, On the ontological expressiveness of information systems analysis and design grammars, Inf. Syst. J. 3 (4) (1993) 217–237.
[87] R. Lukyanenko, V.C. Storey, O. Pastor, Foundations of information technology based on bunge's systemist philosophy of reality, Softw. Syst. Model. 20 (1) (2021) 921–938, Springer.
[88] R. Lukyanenko, R. Weber, A realist ontology of digital objects and digitalized systems, in: Digital First Era — a Joint AIS SIGSAND/SIGPrag Workshop, Virtual Workshop, 2022, pp. 1–5.
[89] L. Balmelli, An overview of the systems modeling language for products and systems development, J. Object Technol. 6 (6) (2007) 149–177, Citeseer.
[90] S. Friedenthal, A. Moore, R. Steiner, A Practical Guide to SysML: The Systems Modeling Language, Morgan Kaufmann, 2014.
[91] P. Fettke, How conceptual modeling is used, Commun. Assoc. Inf. Syst. 25 (1) (2009) 43.
[92] D. Dori, H. Sillitto, What is a system? An ontological framework, Syst. Eng. 20 (3) (2017) 207–219, Wiley Online Library.
[93] M.A. Bunge, Finding Philosophy in Social Science, Yale University Press, New Haven, CT, 1996.
[94] P.A. Corning, The re-emergence of 'emergence': A venerable concept in search of a theory, Complexity 7 (6) (2002) 18–30, Wiley Online Library.
[95] L. von Bertalanffy, General System Theory: Foundations, Development, Applications, Braziller, New York, 1968.
[96] M.A. Bunge, Philosophy of Science: Vol. 2, from Explanation to Justification, Routledge, New York NY, 2017, https://books.google.ca/books?id=NtwzDwAAQBAJ.
[97] P. Checkland, Soft Systems Methodology: A 30 Year Retrospective. Systems Thinking, Wiley, New York NY, 1999.
[98] N. Luhmann, Trust and Power, John Wiley & Sons, New York, 2018.
[99] A. Burton-Jones, E.R. McLean, E. Monod, Theoretical perspectives in IS research: From variance and process to conceptual latitude and conceptual fit, Eur. J. Inf. Syst. 24 (6) (2015) 664–679, Springer.
[100] A.S. Lee, M. Thomas, R.L. Baskerville, Going back to basics in design science: From the information technology artifact to the information systems artifact, Inf. Syst. J. 25 (1) (2015) 5–21, Wiley Online Library.
[101] N. Nan, Capturing bottom-up information technology use processes: A complex adaptive systems model, MIS Quart. (2011) 505–532, JSTOR.
[102] L. von Bertalanffy, J.W. Sutherland, General systems theory: Foundations, developments, applications, IEEE Trans. Syst. Man Cybern. (6) (1974) 592, IEEE.
[103] D. Hammond, The Science of Synthesis: Exploring the Social Implications of General Systems Theory, University Press of Colorado, 2010.
[104] F.E. Kast, J.E. Rosenzweig, General systems theory: Applications for organization and management, Acad. Manage. J. 15 (4) (1972) 447–465, Academy of Management Briarcliff Manor, NY 10510.
[105] C. Mele, J. Pels, F. Polese, A brief review of systems theories and their managerial applications, Serv. Sci. 2 (1–2) (2010) 126–135, INFORMS.
[106] L. Skyttner, General systems theory: Origin and hallmarks, Kybernetes 26 (6) (1996) 16–22, MCB UP Ltd.
[107] R.L. Ackoff, Towards a system of systems concepts, Manage. Sci. 17 (11) (1971) 661–671, INFORMS.
[108] R.L. Ackoff, F.E. Emery, On Purposeful Systems: An Interdisciplinary Analysis of Individual and Social Behavior As a System of Purposeful Events, Transaction Publishers, Piscataway, NJ, 2005.
[109] R.D. Arnold, J.P. Wade, A definition of systems thinking: A systems approach, Procedia Comput. Sci. (44) (2015) 669–678, Elsevier.
[110] J.E. Bailey, Toward a science of metabolic engineering, Science 252 (5013) (1991) 1668–1675.
[111] M.A. Bunge, Treatise on Basic Philosophy: Ontology II: A World of Systems, Reidel Publishing Company, Boston, MA, 1979.
[112] B. Henderson-Sellers, Why philosophize; why not just model? in: Conceptual Modeling, Springer, 2015, pp. 3–17.
[113] C. Partridge, C. Gonzalez-Perez, B. Henderson-Sellers, Are Conceptual Models Concept Models? Springer, 2013, pp. 96–105.
[114] Y. Wand, D.E. Monarchi, J. Parsons, C.C. Woo, Theoretical foundations for conceptual modelling in information systems development, Decis. Support Syst. 15 (4) (1995) 285–304.
[115] M.A. Bunge, Neutron-proton scattering at 8 · 8 and 13 MeV, Nature 156 (3958) (1945) 301, Nature Publishing Group.
[116] A. Burton-Jones, R. Clarke, K. Lazarenko, R. Weber, Is use of optional attributes and associations in conceptual modeling always problematic? Theory and empirical tests, in: International Conference on Information Systems, 2013, pp. 1–14.
[117] O. Pastor, J.C. Molina, Model-Driven Architecture in Practice: A Software Production Environment Based on Conceptual Modeling, Springer Science & Business Media, 2007.
[118] Y. Wand, R. Weber, Mario bunge's ontology as a formal foundation for information systems concepts, in: P. Weingartner, G. Dorn (Eds.), Studies on Mario Bunge's Treatise, Rodopi, New Haven, CT, 1990, pp. 123–150.
[119] Y. Wand, R. Weber, Thirty years later: Some reflections on ontological analysis in conceptual modeling, J. Database Manage. (JDM) 28 (1) (2017) 1–17.
[120] B. Wyssusek, On ontological foundations of conceptual modelling, Scand. J. Inf. Syst. 18 (1) (2006) 63–80.

[121] M.A. Bunge, Between Two Worlds: Memoirs of a Philosopher-Scientist, Springer, Heidelberg, Germany, 2016.
[122] M.A. Bunge, G.M. Denegri, E.L. Ortiz, H.W. Droste, A. Cordero, P. Deleporte, M. Manzano, M.C. Moreno, D. Raynaud, in: M. Matthews (Ed.), Mario Bunge: A Centenary Festschrift, Nature Springer, Cham, Switzerland, 2019.
[123] M.R. Matthews, Mario bunge: An ntroduction to his life, work and achievements, in: Mario Bunge: A Centenary Festschrift, Nature Springer, Cham, Switzerland, 2019, pp. 1–28.
[124] Y. Wand, R. Weber, On the deep-structure of information-systems, Inf. Syst. J. 5 (3) (1995) 203–223.
[125] M.A. Bunge, Treatise on Basic Philosophy: Ontology I: The Furniture of the World, Reidel, Boston, MA, 1977.
[126] M.A. Bunge, Treatise on Basic Philosophy: Ethics: The Good and the Right, Treatise on Basic Philosophy, Springer Netherlands, Amsterdam, 1989, https://books.google.ca/books?id=0zTwCAAAQBAJ.
[127] M.A. Bunge, Chasing Reality: Strife over Realism, University of Toronto Press, 2006.
[128] M.A. Bunge, Treatise on Basic Philosophy: Semantics I: Sense and Reference, Treatise on Basic Philosophy, Springer Netherlands, Amsterdam, 1974, https://books.google.ca/books?id=CGDr1ir6y_8C.
[129] M.A. Bunge, Emergence and Convergence: Qualitative Novelty and the Unity of Knowledge, University of Toronto Press, Toronto, ON, 2003.
[130] M.J. Veltman, Facts and Mysteries in Elementary Particle Physics, World Scientific, New York NY, 2003.
[131] S. Hawking, L. Mlodinow, The Grand Design, Random House Digital, Inc, London, UK, 2010.
[132] M.A. Bunge, Philosophical Dictionary, Prometheus Books, Amherst, NY, 2003.
[133] M.A. Bunge, Philosophy of Science: From Explanation to Justification, Transaction Publishers, New York NY, 1998, https://books.google.ca/books?id=qbEmnwEACAAJ.
[134] J. Gerring, The mechanismic worldview: Thinking inside the box, Br. J. Polit. Sci. 38 (1) (2008) 161–179, Cambridge University Press.
[135] M. Bunge, R. Ardila, Philosophy of Psychology, Springer New York, Berlin/ Heidelberg, 2012, https://books.google.com/books?id=qMe5BgAAQBAJ.
[136] J.L. Austin, J.O. Urmson, M. Sbisà, How to Do Things with Words, in: William James Lectures, Clarendon Press, 1975, https://books.google.com/books?id=XnRkQSTUpmgC.
[137] J.R. Searle, Intentionality: An Essay in the Philosophy of Mind, Cambridge University Press, Cambridge, 1983.
[138] J.R. Searle, The Construction of Social Reality, Simon and Schuster, 1995.
[139] M.K. Chang, C.C. Woo, A speech-act-based negotiation protocol: Design, implementation, and test use, ACM Trans. Inf. Syst. 12 (4) (1994) 360–382, ACM New York, NY, USA.
[140] O. Eriksson, P.J. Agerfalk, Speaking things into existence: Ontological foundations of identity representation and management, Inf. Syst. J. 35 (1) (2021) 1–30, Wiley Online Library.
[141] M.A. Janson, C.C. Woo, Comparing IS development tools and methods: Using speech act theory, Inf. Manage. 28 (1) (1995) 1–12, Elsevier.
[142] M.A. Janson, C.C. Woo, A speech act lexicon: An alternative use of speech act theory in information systems, Inf. Syst. J. 6 (4) (1996) 301–329, Wiley Online Library.
[143] S. March, G. Allen, Toward a social ontology for conceptual modeling, in: 11th Symposium on Research in Systems Analysis and Design, Vancouver, Canada, 2012, pp. 57–62.
[144] R. Dubin, Theory Building, Free Press, 1978, http://books.google.ca/books?id=a0NqAAAAMAAJ.
[145] C. Post, R. Sarala, C. Gatrell, J.E. Prescott, Advancing theory with review articles, J. Manage. Stud. 57 (2) (2020) 351–376, Wiley Online Library.
[146] P. Green, M. Rosemann, M. Indulska, J. Recker, Complementary use of modeling grammars, Scand. J. Inf. Syst. 23 (1) (2011) 59–86, Department of Mathematics and Computer Science, Institute of Electronic ....
[147] M. Hvalshagen, R. Lukyanenko, B.M. Samuel, Empowering users with narratives: Examining the efficacy of narratives for understanding data-oriented conceptual models, Inf. Syst. Res. (2023) 1–38, forthcoming.
[148] M. Jabbari, J. Recker, P. Green, K. Werder, How do individuals understand multiple conceptual modeling scripts? J. Assoc. Inf. Syst. 23 (4) (2022) 1037–1070.
[149] M. Malinova, J. Mendling, Cognitive diagram understanding and task performance in systems analysis and design, MIS Quart. 45 (4) (2021) 2101–2157, Management Information Systems Research Center, Carlson School of Management ....
[150] J. Recker, P. Green, How do individuals interpret multiple conceptual models? A theory of combined ontological completeness and overlap, J. Assoc. Inf. Syst. 20 (8) (2019) 1.
[151] P. Pourali, J.M. Atlee, An empirical investigation to understand the difficulties and challenges of software modellers when using modelling tools, in: Proceedings of the 21th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, 2018, pp. 224–234.
[152] J. Parsons, Y. Wand, Attribute-based semantic reconciliation of multiple data sources, J. Data Semant. (2800) (2003) 21–47.
[153] Y. Wand, R. Weber, An ontological model of an information-system, IEEE Trans. Softw. Eng. 16 (11) (1990) 1282–1292.
[154] R.R. Nelson, Project retrospectives: Evaluating project success, failure, and everything in between, MIS Quart. Exec. 4 (3) (2008) 5.
[155] R.R. Nelson, IT project management: Lessons learned from project retrospectives 1999–2020, Found. Trends® Inf. Syst. 4 (4) (2021) 275–381, Now Publishers, Inc.
[156] R. Bonney, J.L. Shirk, T.B. Phillips, A. Wiggins, H.L. Ballard, A.J. Miller-Rushing, J.K. Parrish, Next steps for citizen science, Science 343 (6178) (2014) 1436–1437.
[157] H. Burgess, L. DeBey, H. Froehlich, N. Schmidt, E. Theobald, A. Ettinger, J. HilleRisLambers, J. Tewksbury, J. Parrish, The science of citizen science: Exploring barriers to use as a primary research tool, Biol. Cons. 208 (1) (2017) 1–8.
[158] M. Levy, M. Germonprez, The potential for citizen science in information systems research, Commun. Assoc. Inf. Syst. 40 (1) (2017) 2.
[159] R. Lukyanenko, J. Parsons, Beyond micro-tasks: Research opportunities in observational crowdsourcing, J. Database Manage. (JDM) 29 (1) (2018) 1–22.
[160] L. Fortson, K. Masters, R. Nichol, K. Borne, E. Edmondson, C. Lintott, J. Raddick, K. Schawinski, J. Wallin, Galaxy zoo: Morphological classification and citizen science, Adv. Mach. Learn. Data Min. Astron. (2011) 1–11.
[161] B.L. Sullivan, C.L. Wood, M.J. Iliff, R.E. Bonney, D. Fink, S. Kelling, Ebird: A citizen-based bird observation network in the biological sciences, Biol. Cons. 142 (10) (2009) 2282–2292.
[162] A. Bowser, K. Shilton, E. Warrick, J. Preece, Accounting for privacy in citizen science: ethical research in a context of openness, in: Proc. of CSCW 2017, 2017.
[163] Q. Groom, L. Weatherdon, I.R. Geijzendorffer, Is citizen science an open science in the case of biodiversity observations? J. Appl. Ecol. 54 (2) (2017) 612–617, Wiley Online Library.
[164] R. Vicente-Saez, C. Martinez-Fuentes, Open science now: A systematic literature review for an integrated definition, J. Bus. Res. (88) (2018) 428–436, Elsevier.
[165] M. Woelfle, P. Olliaro, M.H. Todd, Open science is a research accelerator, Nature Chem. 3 (10) (2011) 745–748, Nature Publishing Group.
[166] M.A. Fielden, A.C. Chaulk, K. Bassett, Y.F. Wiersma, M. Erbland, H. Whitney, T.W. Chapman, Aedes japonicus japonicus (Diptera: Culicidae) arrives at the most easterly point in north America, Canad. Entomol. 147 (06) (2015) 737–740.
[167] R. Lukyanenko, J. Parsons, Y. Wiersma, M. Maddah, Expecting the unexpected: Effects of data collection design choices on the quality of crowdsourced user-generated content, MIS Quart. 43 (2) (2019) 634–647.
[168] R. Lukyanenko, J. Parsons, Conceptual modeling principles for crowdsourcing, in: International Workshop on Multimodal Crowdsensing, ACM, Maui, Hawaii, USA, 2012, pp. 3–6.

[169] N. Prestopnik, K. Crowston, Citizen Science System Assemblages: Understanding the Technologies that Support Crowdsourced Science, ACM, 2012, pp. 168–176.
[170] A. Wiggins, R. Bonney, E. Graham, S. Henderson, S. Kelling, G. LeBuhn, R. Litauer, K. Lots, W. Michener, G. Newman, Data management guide for public participation in scientific research, 2013, pp. 1–41.
[171] R. Lukyanenko, J. Parsons, Y. Wiersma, G. Wachinger, B. Huber, R. Meldt, Representing crowd knowledge: Guidelines for conceptual modeling of user-generated content, J. Assoc. Inf. Syst. 18 (4) (2017) http://dx.doi.org/10.17705/1jais.00456.
[172] J. Parsons, An information model based on classification theory, Manage. Sci. 42 (10) (1996) 1437–1453.
[173] R. Lukyanenko, J. Parsons, Y. Wiersma, The IQ of the crowd: Understanding and improving information quality in structured user-generated content, Inf. Syst. Res. 25 (4) (2014) 669–689.
[174] R. Lukyanenko, J. Parsons, Easier crowdsourcing is better: Designing crowdsourcing systems to increase information quality and user participation, in: Design Science Research. Cases, Springer, Berlin/ Heidelberg, 2020.
[175] T. Gura, Citizen science: Amateur experts, Nature 496 (7444) (2013) 259–261.
[176] E. Lewandowski, H. Specht, Influence of volunteer and project characteristics on data quality of biological surveys, Conserv. Biol. 29 (3) (2015) 713–723.
[177] J. Parsons, R. Lukyanenko, Y. Wiersma, Easier citizen science is better, Nature 471 (7336) (2011) 37.
[178] I. Stirling, Polar Bears, University of Michigan Press, Ann Arbor, Michigan, 1998.
[179] G.C. Bowker, S. Timmermans, A.E. Clarke, E. Balka, Boundary Objects and beyond: Working with Leigh Star, MIT Press, Cambridge, Mass, 2016.
[180] N. Jukic, S. Vrbsky, S. Nestorov, A. Sharma, Database Systems: Introduction to Databases and Data Warehouses, Prospect Press, Burlington, VT, 2019, https://books.google.com/books?id=k0vlwQEACAAJ.
[181] M. Daneva, E. Van Der Veen, C. Amrit, S. Ghaisas, K. Sikkel, R. Kumar, N. Ajmeri, U. Ramteerthkar, R. Wieringa, Agile requirements prioritization in large-scale outsourced system projects: An empirical study, J. Syst. Softw. 86 (5) (2013) 1333–1353, Elsevier.
[182] N.B. Moe, D. Šmite, G.K. Hanssen, H. Barney, From offshore outsourcing to insourcing and partnerships: Four failed outsourcing attempts, Empir. Softw. Eng. 19 (5) (2014) 1225–1258, Springer.
[183] S. Sahay, B. Nicholson, S. Krishna, Global IT Outsourcing: Software Development Across Borders, Cambridge University Press, Cambridge, 2003.
[184] D. Harel, Statecharts: A visual formalism for complex systems, Sci. Comput. Program. 8 (3) (1987) 231–274, Elsevier.
[185] E. Segel, J. Heer, Narrative visualization: Telling stories with data, IEEE Trans. Vis. Comput. Graphics 16 (6) (2010) 1139–1148, IEEE.
[186] I. Dissanayake, S. Nerur, R. Singh, Y. Lee, Medical crowdsourcing: Harnessing the 'wisdom of the crowd' to solve medical mysteries, J. Assoc. Inf. Syst. 20 (11) (2019) 4.
[187] J.H. Frost, M.P. Massagli, Social uses of personal health information within PatientsLikeMe, an online patient community: What can happen when patients have access to one another's data, J. Med. Internet Res. 10 (3) (2008).
[188] J. Kallinikos, N. Tempini, Patient data as medical facts: Social media practices as a foundation for medical knowledge creation, Inf. Syst. Res. 25 (4) (2014) 817–833.
[189] P. Wicks, M. Massagli, J. Frost, C. Brownstein, S. Okun, T. Vaughan, R. Bradley, J. Heywood, Sharing health data for better outcomes on PatientsLikeMe, J. Med. Internet Res. 12 (2) (2010).
[190] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, W. Vogels, Dynamo: Amazon's highly available key-value store, in: ACM SIGOPS Operating Systems Review, Vol. 41, ACM, 2007, pp. 205–220.
[191] S. Idreos, M. Callaghan, Key-value storage engines, in: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, 2020, pp. 2667–2672.
[192] G. Gigerenzer, P.M. Todd, Simple Heuristics that Make us Smart, Oxford University Press USA, New York, NY, 1999.
[193] T. Gilovich, D. Griffin, D. Kahneman, Heuristics and Biases: The Psychology of Intuitive Judgment, Cambridge University Press, Cambridge, UK, 2002.
[194] H. Markovits, R. Vachon, Conditional reasoning, representation, and level of abstraction, Dev. Psychol. 26 (6) (1990) 942.
[195] D. Sherrington, S. Kirkpatrick, Solvable model of a spin-glass, Phys. Rev. Lett. 35 (26) (1975) 1792, APS.
[196] S.N. Ethier, T.G. Kurtz, Markov Processes: Characterization and Convergence, John Wiley & Sons, New York NY, 2009.
[197] J.R. Galbraith, Designing Organizations: Strategy, Structure, and Process At the Business Unit and Enterprise Levels, Wiley, New Delhi, 2014, https://books.google.ca/books?id=KVd5AgAAQBAJ.
[198] B. Kitchens, S.L. Johnson, P. Gray, Understanding echo chambers and filter bubbles: The impact of social media on diversification and partisan shifts in news consumption, MIS Quart. 44 (4) (2020).
[199] N. Levina, M. Arriaga, Distinction and status production on user-generated content platforms: Using Bourdieu's theory of cultural production to understand social dynamics in online fields, Inf. Syst. Res. 25 (3) (2014) 468–488.
[200] A. Susarla, J.-H. Oh, Y. Tan, Social networks and the diffusion of user-generated content: Evidence from YouTube, Inf. Syst. Res. 23 (1) (2012) 23–41.
[201] O. Arazy, O. Nov, R. Patterson, L. Yeo, Information quality in wikipedia: The effects of group composition and task conflict, J. Manage. Inf. Syst. 27 (4) (2011) 71–98.
[202] O. Arazy, O. Nov, Determinants of wikipedia quality: The roles of global and local contribution inequality, in: Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work, ACM, Savannah, Georgia, USA, 2010, pp. 233–236.
[203] S. Hansen, N. Berente, K. Lyytinen, Wikipedia as rational discourse: An illustration of the emancipatory potential of information systems, in: System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on, IEEE, 2007, p. 253.
[204] E. Björgvinsson, P. Ehn, P.-A. Hillgren, Agonistic participatory design: Working with marginalised social movements, CoDesign 8 (2–3) (2012) 127–144.
[205] T. Bratteteig, I. Wagner, Disentangling Participation: Power and Decision-Making in Participatory Design, Springer International Publishing, New York, NY, 2014.
[206] Y. Fang, D. Neufeld, Understanding sustained participation in open source software projects, J. Manage. Inf. Syst. 25 (4) (2009) 9–50, Taylor & Francis.
[207] N. Iivari, Participatory design in OSS development: Interpretive case studies in company and community OSS development contexts, Behav. Inf. Technol. 30 (3) (2011) 309–323.
[208] R. van Wendel de Joode, Managing conflicts in open source communities, Electron. Mark. 14 (2) (2004) 104–113, Taylor & Francis.
[209] M. Albert, V. Pelechano, J. Fons, M. Ruiz, O. Pastor, Implementing UML Association, Aggregation, and Composition. A Particular Interpretation Based on a Multidimensional Framework, Springer, 2003, pp. 143–158.
[210] V.C. Storey, Meronymic relationships, J. Database Manage. (JDM) 2 (3) (1991b) 22–36, IGI Global.
[211] G. Guizzardi, H.A. Proper, On understanding the value of domain modeling, in: Proceedings of 15th International Workshop on Value Modelling and Business Ontologies (VMBO 2021), 2021.
[212] M. Op't Land, E. Proper, M. Waage, J. Cloo, C. Steghuis, The results of enterprise architecting, in: Enterprise Architecture, Springer, 2009, pp. 49–83.
[213] L.C. Briand, Y. Labiche, J. Cui, Automated support for deriving test requirements from UML statecharts, Softw. Syst. Model. 4 (4) (2005) 399–423, Springer.
[214] I. Habli, W. Wu, K. Attwood, T. Kelly, Extending Argumentation to Goal-Oriented Requirements Engineering, Springer, 2007, pp. 306–316.
[215] J. Mylopoulos, Conceptual modeling and telos, in: P. Loucopoulos, R. Zicari (Eds.), Conceptual Modeling, Databases, and CASE: An Integrated View of Information Systems Development, John Wiley & Sons, Inc, New York, NY, 1992, pp. 49–68.
[216] E. Paja, A. Maté, C. Woo, J. Mylopoulos, Can Goal Reasoning Techniques Be Used for Strategic Decision-Making?, Springer, 2016, pp. 530–543.

[217] A. Burton-Jones, P.N. Meso, The effects of decomposition quality and multiple forms of information on novices' understanding of a domain from a conceptual model, J. Assoc. Inf. Syst. 9 (12) (2008) 748–802.
[218] M. Hvalshagen, B.M. Samuel, R. Lukyanenko, Conceptual data models and narratives: A tool to help the tool, in: SIGSAND Symposium, Cincinnati, Ohio, 2017, pp. 1–10.
[219] S. Bandini, S. Manzoni, G. Vizzari, Agent based modeling and simulation: An informatics perspective, J. Artif. Soc. Soc. Simul. 12 (4) (2009) 4.
[220] S.F. Railsback, V. Grimm, Agent-Based and Individual-Based Modeling: A Practical Introduction, Princeton University Press, Princeton, NJ, 2019.
[221] K.E. Weick, Theory construction as disciplined imagination, Acad. Manag. Rev. 51 (1989) 6–531.
[222] K.E. Weick, What theory is not, theorizing is, Administrative Science Quarterly 40 (3) (1995) 385–390.
[223] M.A. Bedau, Weak emergence, Philos. Perspect. (11) (1997) 375–399, JSTOR.
[224] A. Saghafi, Y. Wand, J. Parsons, Skipping class: Improving human-driven data exploration and querying through instances, Eur. J. Inf. Syst. (2021) 1–29, Taylor & Francis.
[225] B.M. Samuel, V. Khatri, V. Ramesh, Exploring the effects of extensional versus intentional representations on domain understanding, MIS Quart. 42 (4) (2018) 1187–1209.
[226] R.C. Goldstein, V.C. Storey, Data abstractions: Why and how? Data Knowl. Eng. 29 (3) (1999) 293–311, Elsevier.
[227] R. Motschnig-Pitrik, V.C. Storey, Modelling of set membership: The notion and the issues, Data Knowl. Eng. 16 (2) (1995) 147–185, Elsevier.
[228] A. Olivé, Conceptual Modeling of Information Systems, Springer Science & Business Media, Berlin, Germany, 2007.
[229] S. Ali, T. Yue, R. Abreu, When software engineering meets quantum computing, Commun. ACM 65 (4) (2022) 84–88, ACM New York, NY, USA.
[230] A. Seiden, Particle Physics: A Comprehensive Introduction, Addison-Wesley, 2005.
[231] S.F. Aikin, Who is afraid of epistemology's regress problem? Philos. Stud. 126 (2) (2005) 191–217, Springer.
[232] R.L. Bliss, Viciousness and the structure of reality, Philos. Stud. 166 (2) (2013) 399–418, Springer.
[233] R.P. Cameron, Turtles all the way down: Regress, priority and fundamentality, Philos. Quart. 58 (230) (2008) 1–14, Blackwell Publishers Ltd Oxford, UK and Boston, USA.
[234] D. Nolan, What's wrong with infinite regresses? Metaphilosophy 32 (5) (2001) 523–538, Wiley Online Library.
[235] J.J. Smart, The river of time, Mind 58 (232) (1949) 483–494, JSTOR.
[236] D. Batra, Cognitive complexity in data modeling: Causes and recommendations, Requir. Eng. 12 (4) (2007) 231–244.
[237] M. Kaul, V.C. Storey, C. Woo, A framework for managing complexity in information systems, J. Database Manage. 28 (1) (2017) 31–42, IGI Global.
[238] D.L. Moody, The 'physics' of notations: Toward a scientific basis for constructing visual notations in software engineering, Softw. Eng. IEEE Trans. 35 (6) (2009) 756–779.
[239] H. Topi, V. Ramesh, Human factors research on data modeling: A review of prior research, an extended framework and future research directions, J. Database Manage. 13 (2) (2002) 3–19.
[240] A. Burton-Jones, Y. Wand, R. Weber, Guidelines for empirical evaluations of conceptual modeling grammars, J. Assoc. Inf. Syst. 10 (6) (2009) 495–532.
[241] R. Lukyanenko, J. Parsons, B.M. Samuel, Artifact sampling in experimental conceptual modeling research, in: EmpER'18 — First International Workshop on Empirical Methods in Conceptual Modeling, Xi'an, China, 2018.
[242] J. Parsons, L. Cole, What do the pictures mean? Guidelines for experimental evaluation of representation fidelity in diagrammatical conceptual modeling techniques, Data Knowl. Eng. 55 (3) (2005) 327–342.
[243] R.E. Mayer, Multimedia learning, in: Psychology of Learning and Motivation, Vol. 41, Elsevier, 2002, pp. 85–139.
[244] R.E. Mayer, R. Moreno, Nine ways to reduce cognitive load in multimedia learning, Educ. Psychol. 38 (1) (2003) 43–52.
[245] H. Avdiji, D. Elikan, S. Missonier, Y. Pigneur, A design theory for visual inquiry tools, J. Assoc. Inf. Syst. 21 (3) (2020) 1–34.
[246] A. Osterwalder, Y. Pigneur, Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers, John Wiley & Sons, New York NY, 2010.
[247] A. Osterwalder, Y. Pigneur, Designing business models and similar strategic objects: The contribution of IS, J. Assoc. Inf. Syst. 14 (5) (2012) 3.
[248] P. Kroes, Technical Artefacts: Creations of Mind and Matter: A Philosophy of Engineering Design, Vol. 6, Springer Science & Business Media, 2012.
[249] R. Lukyanenko, J. Parsons, Design theory indeterminacy: What is it, how can it be reduced, and why did the polar bear drown? J. Assoc. Inf. Syst. 21 (5) (2020) 1–30.
[250] D.S. Smyth, .P.B. Checkland, Using a systems approach: The structure of root definitions, J. Appl. Syst. Anal. 5 (1) (1976) 75–83.
[251] A. Basden, A.T. Wood-Harper, A philosophical discussion of the root definition in soft systems thinking: An enrichment of CATWOE, Syst. Res. Behav. Sci. Off. J. Int. Fed. Syst. Res. 23 (1) (2006) 61–87, Wiley Online Library.
[252] B. Bergvall-Kåreborn, A. Mirijamdotter, A. Basden, Basic principles of SSM modeling: An examination of CATWOE from a soft perspective, Syst. Pract. Action Res. 17 (2) (2004) 55–73, Springer.
[253] A. Gemino, Y. Wand, Empirical comparison of object-oriented and dataflow models, in: Proceedings of the Eighteenth International Conference on Information Systems, Association for Information Systems, 1997, pp. 446–447.
[254] Y.-G. Kim, S.T. March, Comparing data modeling formalisms, Commun. ACM 38 (6) (1995) 103–115.
[255] J. Recker, M. Rosemann, P. Green, M. Indulska, Do ontological deficiencies in modeling grammars matter? MIS Quart. 35 (1) (2011) 57–79.
[256] M. Verdonck, F. Gailly, R. Pergl, G. Guizzardi, B. Martins, O. Pastor, Comparing traditional conceptual modeling with ontology-driven conceptual modeling: An empirical study, Inf. Syst. (81) (2019) 92–103.
[257] P. Bera, P. Soffer, J. Parsons, Using eye tracking to expose cognitive processes in understanding conceptual models, MIS Quart. 43 (4) (2019) 1105–1126.
[258] L.M. Delcambre, S.W. Liddle, O. Pastor, V.C. Storey, Articulating Conceptual Modeling Research Contributions, in: Springer, 2021, pp. 45–60.
[259] S. Purao, V.C. Storey, A multi-layered ontology for comparing relationship semantics in conceptual models of databases, Appl. Ontol. 1 (1) (2005) 117–139, IOS Press.
[260] A. Saghafi, Y. Wand, Conceptual models? A meta-analysis of empirical work, in: Hawaii International Conference on System Sciences, Big Island, HI, 2014, pp. 1–15.
[261] K. Siau, M. Rossi, Evaluation of Information Modeling Methods-a Review, Vol. 5, IEEE, 1998, pp. 314–322.
[262] E. Söderström, B. Andersson, P. Johannesson, E. Perjons, B. Wangler, Towards a framework for comparing process modelling languages, in: International Conference on Advanced Information Systems Engineering, Springer, 2002, pp. 600–611.
[263] P. Domingos, The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World, Basic Books, New York NY, 2015, https://books.google.com/books?id=WpTSDQAAQBAJ.
[264] D. Bork, Conceptual modeling and artificial intelligence: Challenges and opportunities for enterprise engineering, in: Enterprise Engineering Working Conference, Springer, 2022, pp. 3–9.
[265] P. Fettke, Conceptual modelling and artificial intelligence: Overview and research challenges from the perspective of predictive business process management, 2020, pp. 157–164.
[266] R. Lukyanenko, A. Castellanos, J. Parsons, M.C. Tremblay, V.C. Storey, Using conceptual modeling to support machine learning, in: International Conference on Advanced Information Systems Engineering, Springer, 2019, pp. 170–181.
[267] W. Maass, V.C. Storey, Pairing conceptual modeling with machine learning, Data Knowl. Eng. (2021) 101–123, Elsevier.
[268] J. Evermann, Y. Wand, Ontological modeling rules for UML: An empirical assessment, J. Comput. Inf. Syst. (46) (2006) 14–29.

[269] G. Guizzardi, G. Wagner, What's in a relationship: An ontological analysis, in: International Conference on Conceptual Modeling, Springer, 2008, pp. 83–97.
[270] I. Hadar, P. Soffer, Variations in conceptual modeling: Classification and ontological analysis, J. Assoc. Inf. Syst. 7 (8) (2006) 20.
[271] I. Reinhartz-Berger, A. Sturm, Y. Wand, Comparing functionality of software systems: An ontological approach, Data Knowl. Eng. (87) (2012) 320–338.
[272] T.P. Sales, N. Guarino, G. Guizzardi, J. Mylopoulos, An ontological analysis of value propositions, in: EDOC, IEEE, 2017, pp. 184–193.
[273] Y. Wand, Using object concepts and UML for conceptual modeling, Adv. Conceptual Model. Chall. Oppor. (2008) 101–102.
[274] C. Welty, N. Guarino, Supporting ontological analysis of taxonomic relationships, Data Knowl. Eng. 39 (1) (2001) 51–74, Elsevier.
[275] S.W. Ambler, Agile model driven development is good enough, IEEE Softw. 20 (5) (2003) 71–73, IEEE.
[276] S. Beydeda, M. Book, V. Gruhn, Model-Driven Software Development, Vol. 15, Springer, Berlin/ Heidelberg, 2005.
[277] X. Yu, Y. Zhang, T. Zhang, L. Wang, J. Hu, J. Zhao, X. Li, A model-driven development framework for enterprise web services, Inf. Syst. Front. 9 (4) (2007) 391–409, Springer.
[278] C. Giannoulis, J. Zdravkovic, Modeling strategy maps and balanced scorecards using istar, 2011, pp. 90–95.
[279] J. Horkoff, S. Eric, Qualitative, interactive, backward analysis of I* models, 2008, pp. 43–46.

Roman Lukyanenko is an associate professor at the University of Virginia. Roman obtained his Ph.D. from Memorial University of Newfoundland under the supervision of Jeffrey Parsons. Roman's research interests include conceptual modeling, information quality, crowdsourcing, machine learning, design science research, and research methods (research validities, instantiation validity, and artifact sampling). Roman's work on these topics has been published in *Nature*, *MIS Quarterly*, *Information Systems Research*, *Journal of the Association for Information Systems*, *European Journal of Information Systems*, among others. His 2019 paper on quality of crowdsourced data received the Best Paper Award at *MIS Quarterly* and was named one of best information systems papers of the year by the AIS College of Senior Scholars. Roman served as a president of AIS SIGSAND and is a co-developer of www.sigsand.com.

Veda C. Storey is the Tull Professor of Computer Information Systems and professor of computer science at the J. Mack Robinson College of Business, Georgia State University. Her research interests are in intelligent information systems, data management, conceptual modeling, and design science research. She is particularly interested in the assessment of the impact of new technologies on business and society from a data management perspective. She is a member of AIS College of Senior Scholars and the steering committee of the International Conference of Conceptual Modeling.

Oscar Pastor is Full Professor and Director of the Research Center on"Métodos de Producción de Software (PROS)" at the Universidad Politécnica de Valencia (Spain). He received his Ph.D. in 1992, Supervisor of 32 Ph.D. Thesis and with an h-index of 49 according to Google Scholar, he has published more than four hundred research papers in conference proceedings, journals and books, and received numerous research grants from public institutions and private industry. Member of the SC of conferences as ER, CAiSE, ESEM, REFSQ, EICS, ICWE, CIbSE or RCIS, his research activities focus on conceptual modeling, web engineering, requirements engineering, information systems design, and model-based software production. He created the object-oriented, formal specification language OASIS and the corresponding software production method OO-METHOD, based on which he led the research and development underlying CARE Technologies that was formed in 1996, creating an advanced MDA-based Conceptual Model Compiler called IntegraNova. Recipient of the Peter Chen Award on Conceptual Modeling in 2016, he is currently leading a multidisciplinary project linking Information Systems and Bioinformatics notions, oriented to design and implement tools for Conceptual Modeling-based interpretation of the Human Genome.