



Universal conceptual modeling: principles, benefits, and an agenda for conceptual modeling research

Roman Lukyanenko¹ · Binny M. Samuel² · Jeffrey Parsons³ · Veda C. Storey⁴ · Oscar Pastor⁵ · Araz Jabbari⁶

Received: 11 November 2023 / Revised: 26 June 2024 / Accepted: 5 August 2024
© The Author(s) 2024

Abstract

The paper proposes *universal conceptual modeling*, conceptual modeling that strives to be as general-purpose as possible and accessible to anyone, professionals and non-experts alike. The idea of universal conceptual modeling is meant to catalyze new thinking in conceptual modeling and be used to evaluate and develop conceptual modeling solutions, such as modeling languages, approaches for requirements elicitation, or modeling tools. These modeling solutions should be usable by as many people and design agents as possible and for as many purposes as possible, aspiring to the ideals of universal conceptual modeling. We propose foundations of universal conceptual modeling in the form of six principles: flexibility, accessibility, ubiquity, minimalism, primitivism, and modularity. We then demonstrate the utility of these principles to evaluate existing conceptual modeling languages and understand conceptual modeling practices. Finally, we propose future research opportunities meant to realize the ideals of universal conceptual modeling.

Keywords Universal conceptual modeling · Conceptual modeling · Conceptual modeling foundations · Universal modeling language · Datish · RDF · Inclusive modeling · Universal design

1 Introduction

Traditionally, conceptual modeling languages,¹ methods and tools were tailored to specific applications and targeted specific groups of users (mainly IT professionals). Thus,

Communicated by Dominik Bork and Henderik Proper.

✉ Roman Lukyanenko
romanl@virginia.edu

Binny M. Samuel
samuelby@uc.edu

¹ McIntire School of Commerce, University of Virginia, Charlottesville, USA

² Lindner College of Business, University of Cincinnati, Cincinnati, USA

³ Faculty of Business Administration, Memorial University of Newfoundland, St. John's, Canada

⁴ Georgia State University, Atlanta, USA

⁵ Universitat Politècnica de Valencia, València, Spain

⁶ Faculty of Business Administration, Université Laval Pavillon Palasis-Prince, 2325 Rue de la Terrasse, Québec, QC G1V 0A6, Canada

¹ Table A1 in the appendix provides the definitions of key terms used in the paper.

Thalheim [111, 161, p. 2] observes, “there is no universal [conceptual modeling] approach and no universal language.” Ambler [3, p. 47] explains that “[b]ecause each type of model has its strengths and weaknesses, no single model is sufficient for your modeling needs.” However, a general-purpose, universal conceptual modeling approach warrants consideration due to 1) the ubiquity of information technology; 2) increased systems development and use by non-IT professionals, employees and members of the general public; and 3) continued proliferation of different modeling approaches. Furthermore, as new applications of information technology continually emerge, the need to integrate different systems and provide common modeling solutions becomes more urgent.

Consider a common scenario in which an organization seeks to communicate to offshore agile developers the requirements for a new information system [1, 74]. Conceptual modeling is well-positioned to facilitate this process. However, many organizational employees struggle to use conceptual models, in part due to the steep learning curve [75]. Meanwhile, agile developers shun formal conceptual modeling seeing it as wasteful overhead [8]. In the words of Ron Jeffries, a co-author of Agile Manifesto, “software

Table 1 Existing advances to universal conceptual modeling (UCM)

Prior efforts related to UCM	Unaddressed UCM requirements
<ul style="list-style-type: none"> ·Increased Expressiveness (e.g., Extended ER) ·More Precise Semantics (e.g., OntoUML) ·Less Technical (IT) Audience to promote accessibility (e.g., ConML, FlexiSketch) ·New Domains (e.g., enterprise modeling such as domain modeling, ArchiMate, or SysML, organizational systems such as OO-Method, or engineering tasks such as SysML) ·Constructs based on principles of visual notation 	<ul style="list-style-type: none"> ·Objects belong to predefined entities/classes ·Still anchored in IT concepts for IT audience ·High model complexity/learning curve (e.g., number of constructs) ·Constrained domains/contexts ·Grounded in specific problems and not theory ·No universal shapes, constructs, or construct names

development...is best done with as little modeling as possible" (quoted in [3], p. xi). Despite agile emerging as a dominant development approach, only around 40% of agile projects are successful [80]. The causes of failure include communication breakdowns and faulty requirements (especially in large, distributed, and non-located projects) [80]. A simple, easy to learn and use conceptual modeling language that can be used by developers and users can greatly enhance communication and requirements analysis. Models created directly by the users can also alleviate the burden on agile developers to be the sole modelers—a key obstacle to modeling adoption in agile settings [3].

This paper proposes foundations and a framework for *universal conceptual modeling*. Universal conceptual modeling (UCM) is conceptual modeling that is: (1) general-purpose, meaning it can be used to model any domain, and (2) generally usable, meaning it can be used and understood by the broadest audiences possible with very minimal or even no training requirements. Hence, UCM is an ideal type of conceptual modeling that can serve as a frame of reference or comparison, akin to frictionless motion, free market, or Turing machines. It is a systematically laid out aspiration for the development of conceptual modeling that seeks to accommodate as many people, intelligent agents, and modeling scenarios as possible.

The ideal of universal conceptual modeling and the principles of UCM are meant to catalyze new thinking in conceptual modeling and can be used to evaluate and develop conceptual modeling solutions, such as a universal modeling language, or a universal approach for requirements elicitation, with corresponding modeling tools. All of these should be usable by as many people as possible and for as many purposes as possible. The benefits of UCM are expected to accrue for all types of modeling [51, 108], including data, process, goal, and value modeling.

The contribution of this paper is developing the concept of universal conceptual modeling. This benefits conceptual modeling research and practice in two specific ways:

- *Supporting design and evaluation of conceptual modeling solutions.* Equipped with the principles of UCM, it

should be possible for researchers to effectively evaluate existing and develop new conceptual modeling languages, methods, and tools. Because there is an unbounded range of design choices, design principles are needed to make conceptual modeling universal and accessible. Relying on common sense when making design decisions may not result in the most effective designs. We thus contribute to the now widely accepted practice in conceptual modeling of providing a theoretical basis for design solutions.

- *Facilitating research and practice of inclusive conceptual modeling.* An important facet of UCM is ensuring that many people can engage in conceptual modeling. This makes UCM relevant to the recent calls for greater inclusion into conceptual modeling of people with different backgrounds, genders, abilities, beliefs [95, 100, 120, 143]. Universal conceptual modeling provides a specific framework to advance future research efforts to make conceptual modeling more inclusive by focusing on the features of universal and, therefore, more accessible and accommodating conceptual modeling.

In this paper, we provide the foundations of universal conceptual modeling in the form of universal conceptual modeling principles. We evaluate several existing conceptual modeling languages with respect to these principles. Since no existing conceptual modeling language fully adheres to the proposed principles, we also illustrate how these principles can be applied in developing a new language. We propose a potential universal data modeling language *Datish* (as in English or Spanish). We show how the principles of UCM provide guidance and rationale for the design of the constructs and rules of *Datish* such that this language can be “spoken” by anyone in any modeling situation; that is, anyone should be able to produce, interpret and use models in *Datish*. Equipped with theoretically sound design principles, the constructs, and rules of *Datish* can be developed in a more rigorous manner and with greater transparency, thereby increasing confidence in the usability and expressive power of *Datish*.

Finally, to support future research guided by UCM, we outline an agenda for conceptual modeling scholarship to

extend and apply these principles. The remainder of this paper is organized following these contributions.

2 Background: rationale and past approaches to universal conceptual modeling

2.1 Expected benefits of universal conceptual modeling

There are significant potential benefits to making conceptual modeling more universal, thereby addressing growing challenges faced by modelers, IT developers, and diverse users of conceptual models. Below, we identify existing unresolved conceptual modeling challenges from an analysis of the conceptual modeling literature.

Modeling for everyone. An important societal trend is involving non-IT employees and members of the public in data modeling tasks. The assumption behind universal conceptual modeling is the ability of as many people as possible to directly (independent of IT professionals) create or read conceptual modeling diagrams and use conceptual modeling tools. These diagrams could be created by IT professionals as well, but there should be very little (*ideally* no) training required to read them. Hence, while diverse stakeholders have been routinely involved in conceptual modeling [41, 46, 51], they were not expected to lead this process or do it on their own. Increasingly, however, non-IT professionals take charge of IT design and data use and require more intuitive, easy to use, accessible conceptual modeling solutions to support their work [75, 131].

Traditional modeling approaches are not intended to support IT non-experts [75]. For example, many common conceptual modeling languages are complex and difficult to comprehend [40, 111]. Often modelers simplify complex languages by partially using them or adding complementary constructs [111]. Domain-specific modeling languages assist non-experts with intuitive frameworks, but still rely on the modelers' domain knowledge and expertise to utilize predefined components and integrate custom elements effectively [53]. The success of these languages relies on the modelers' ability to connect simplified constructs with domain-specific complexities.

While it is natural to think that non-experts would struggle to model using traditional modeling techniques, an often-overlooked fact is that even IT professionals may not be as familiar or comfortable using the traditional modeling techniques. Especially in agile settings where often little formal modeling is practiced, “many” developers themselves are unfamiliar with the intricate syntax of traditional conceptual modeling and “avoid learning modeling” all together [3, p. 90]. This, as Ambler fears, can lead to “devastating”

consequences (*ibid*), as faulty systems endanger the safety and well-being of people and degrade the environment.

More accessible modeling can benefit specific communities among IT non-experts and IT professionals. Presently, conceptual modeling languages struggle to support people (IT professionals or not) with disabilities, marginalized communities, and the elderly [95, 100, 143, 148, 155]. Few sound solutions are provided for them, and only recently has a dialog on this topic been started [95, 100, 143]. Our proposal for universal conceptual modeling—both as a call for action and as a set of concrete principles—seeks to address these vital emerging needs in our society.

Modeling emerging systems and domains. Traditional conceptual modeling languages were not designed to support emerging data-intensive applications such as AI, social media, NoSQL, and data lakes [15, 99, 115]. For example, given the range of databases (e.g., relational, key-value pair, graph-based, columnar, hybrid, spatial, temporal), a pressing practical challenge is how to select an appropriate storage technology. A universal modeling language can help understand the general patterns in a domain (e.g., prevalence of unique objects vs highly similar objects), supported by an appropriate method for analyzing such patterns, which can help guide analysts toward a suitable storage approach.

Another benefit of universal modeling is in being flexible in what and how to model. Many popular approaches to development, such as Agile or DevOps, forego formal modeling, partially due to misalignment between the values of heavyweight traditional a priori modeling and the more lightweight informal modeling needed in these settings [3, 8, 87].

Universal conceptual modeling should support emerging and informal modeling efforts, such as those common in agile settings. For example, consider a scenario where (non-technical) organizational employees wish to help agile developers by sketching their requirements for the new system. The present solutions to agile modeling (e.g., [3]) continue to be based on traditional modeling techniques (e.g., UML diagrams, entity-relationship diagrams, Class-responsibility-collaboration cards), which require significant familiarity with modeling and remain challenging for non-technical users to use. Furthermore, these popular conceptual modeling languages are built on the assumption that the resulting models should be “accurate and complete” [122]. They have limited ability to capture very crude, imperfect, patchy and emerging understanding of systems, features, actors, resources, organizations, tools, policies, and the broader environment.

Universal conceptual modeling is meant to be applicable in all situations, including in situations where the assumptions of accuracy, completeness and familiarity with formal modeling techniques do not hold. In the same vein, an informal, “lightweight” modeling approach can be helpful for

early stages of data exploration and analysis [3, 29, 83, 89]. Hence, it has the potential to better support informal modeling approaches taken by agile. Indeed, flexibility in modeling features is an emerging goal for the conceptual modeling community [108].

Broader systems and data integration. Many modern technologies have multiple components (e.g., an AI-based ad recommender running on a social media platform powered by a highly scalable distributed database). A language capable of modeling all elements of the system could be especially valuable for systems that integrate data across an organization.

Even though there are approaches for modeling such systems, they all eschew the complexity of having to represent domains in their full diversity. A final, unified view is shown instead (e.g., DataVault approach) [59]. Hence, modeling that take a holistic approach (e.g., integrates systems, processes, and data) has been suggested as a key opportunity for future research and development [108].

Having a universal language and method might make it possible to model individual data sources and their interconnections simultaneously using a single language. This would help to better support the development and use of integrated technologies and promote broader understanding of data and systems in the entire enterprise.

Taming complexity of modeling languages. In addition to the emergence of new languages, existing languages continue to expand, seeking greater expressiveness. For example, UML contains 14 different types of diagrams, each with specialized elements, yet practitioners utilize only a fraction of these elements [46, 51]. However, little effort has been undertaken to theoretically distill a “universal core” of conceptual modeling [2]. Having this core is important, given that ordinary people, such as functional employees or members of the public, are beginning to conduct conceptual modeling. It is unreasonable to expect them to be able to filter out non-core elements in a complex language. The presence of more advanced features (e.g., participation cardinality) has been shown to impede domain understanding by non-experts [75]. This is consistent with the findings on negative impacts of technologies with too many features or *bloatware* [49]. For novices, many modeling languages are bloated.

Furthermore, multiple conceptual models may lead to conflicts [2, 138]. If it is possible to express the same domain semantics (e.g., elements, characteristics, rules, meanings, and constraints associated with a specific domain) using a single conceptual modeling diagram, is both efficient and minimizes the potential for error and inconsistencies. Furthermore, a lean language could make it easier to reconcile different representations by different stakeholders under the

assumption that it could be easier to agree on more abstract semantics than on a more specific one.²

The challenges and opportunities of the modern information technology environment potentially can be addressed by more flexible and more universally accessible conceptual modeling. Two factors suggest it is possible to create such a language. First, precedents for a universal language exist in the field of computing, including binary and first-order logic. Second, conceptual modeling languages already rely on common principles of communication, language, and design [95, 100]. The common core of ways in which humans represent reality makes it possible to model diverse, complex, and emerging systems and domains for various people. Hence, the universality of certain aspects of human cognition and communication can be leveraged in distilling the essence of universal conceptual modeling.

2.2 Relevant past approaches to universal conceptual modeling

Many past efforts contribute to the goal of universal conceptual modeling. Some conceptual modeling languages have been widely adopted but cannot be considered “universal.” Notably, the entity-relationship model (ERM) and the Unified Modeling Language (UML) are widely used for database design and software engineering, respectively. There have been extensions of these languages to make their semantics more precise and accessible outside of software engineering (e.g., ConML, [61]). Other relevant efforts include Domain Modelling [12], ArchiMate [85], or SysML [90].

Effectively, UCM seeks to reconcile what may appear to be two opposite goals: the ability to model as much of the real world as possible, while at the same time being as easy to learn and use as possible. In an ideal case, universal conceptual modeling should without restriction include and enable *anyone* to model *anything*. Existing approaches to conceptual modeling do not satisfy these requirements, as discussed below and summarized in Table 1.

Wide-scope modeling. Several established conceptual data modeling languages are used for a wide range of applications, with the ER model and UML class diagrams de facto standards for relational database design and software engineering, respectively. Extensions of these languages make them more expressive (e.g., extended ER model [153, 158]) and their semantics more precise (e.g., OntoUML, [67]) and accessible (e.g., ConML, [61]). Building on these, other

² This follows from cognitive theories of thinking and classification. More abstract, higher-level categories partition the world into fewer subsets and commonly require less specialized knowledge [112]. For example, it is generally easier to agree that a particular person is a *customer*, rather than whether the person is a *loyal* customer.

wide-applicability efforts include enterprise modeling languages [62], such as Domain Modelling [12], ArchiMate [6], or SysML [90].

Despite their wide applicability, established languages do not meet our requirements for a universal language. First, they have known limitations for some modeling tasks. Notably, a common feature of these languages is the *assumption of inherent classification*—requiring or encouraging objects to be members of predefined entity types or classes [126]. This makes it challenging to use these languages to support evolving requirements and heterogeneous data requirements in artificial intelligence, analytics, social media, and NoSQL contexts [4, 50, 98, 99].

Second, these languages are often geared toward a technical audience, rather than novices or members of the public. Some even require highly advanced technical skills. This is, for example, the explicit target audience of Bjørner’s domain engineering approach [12]. Another example is a “unified” approach proposed by Al-Fedaghi [2]. However, it is unclear whether this approach that blends substance and process ontology is indeed accessible to IT professionals, let alone to IT novices. Finally, as already noted, many general-scope languages are quite complex, making their learning and use curve quite steep for non-expert modelers.

Third, some of the approaches have a wide, but still a constrained domain, given the emerging modeling use cases. For example, enterprise modeling offers solutions for streamlining production processes, policy development, resource allocation, and sustainable practices, among others. However, these applications are defined within organizational contexts, especially large ones [54, 88]. To handle the complexity of large organizations, modelers often use languages partially or add complementary constructs, as needed. However, many applications of conceptual modeling extend beyond typical organizational contexts, including areas such as the human genome [129], online citizen science [101], or games [161]. There is also an emerging area of citizen modeling, whereby ordinary people use formal and informal conceptual modeling to support their data work as well as information technology related activities. All these examples do not align with the intent of enterprise modeling but are of interest to universal conceptual modeling.

Ongoing efforts to develop a common language for specific contexts include NoSQL databases or big data [19, 31, 32]. Similarly, the OO-Method [128, 130] is intended for modeling organizational systems, whereas SysML [90] focuses on engineering tasks.

Common language efforts include frameworks, such as RDF/OWL, Petri nets, graph models, and their extensions (e.g., HERAKLIT [52]) that have built-in flexibility. While flexible and applicable to many applications, these approaches do not consider broad audiences and cater to seasoned developers. Also, it is questionable whether extreme

flexibility (e.g., RDF) leads to a high level of construct overload (e.g., when the same element can be used for modeling individuals and categories).

Although the notion of a “universal data language” emerged in industry [170], it seeks to support the development of services, rather than to propose a new conceptual modeling approach. In general, developments in practice are motivated by specific problems (e.g., semantic interoperability), and not grounded in underlying theories, making it difficult to understand why a solution is effective and why it is expected to work beyond a specific setting.

Modeling for wider audiences. Efforts to make conceptual modeling more accessible to less-technical users include ConML [61], FlexiSketch [172], and Rich Pictures [5] which enable modeling to have new uses and users. However, they have notable restrictions: the inherent classification assumption of ConML, the visual forms focus on FlexiSketch, and the “largely subjective, unique to a particular problem situation” [5] nature of rich pictures. They also cater to modelers with some experience. For example, the “target users of FlexiSketch are: (i) software or systems engineers who create sketches during a system development project and (ii) requirements engineers (business analysts) who use sketches to create and communicate requirements” [172].

Existing approaches to conceptual modeling reveal notable efforts to make modeling accessible, intuitive, and broadly applicable. This need has been acknowledged from the earliest days of modeling research and practice (e.g., [33]). Still, a dedicated and systematic effort to understand what universal conceptual modeling entails is missing. Scattered agile practices eventually led to the development of the Agile manifesto [9] and facilitated the establishment of agile methodologies (e.g., Scrum, Kanban, Extreme Programming). Similarly, we seek to provide a manifesto of universal conceptual modeling, formalized as a set of principles to facilitate further dialog and design efforts.

Prior work has also considered some properties relevant for universal conceptual modeling. Notably, when developing visual notations, Moody [110] proposed several principles: semiotic clarity, perceptual discriminability, semantic transparency, complexity management, cognitive integration, visual expressiveness, dual coding, graphical economy, and cognitive fit. As these principles are instrumental in building scientifically grounded effective visualizations, they are relevant. However, they do not specifically consider the design of universal conceptual modeling. For example, they do not identify universal shapes, construct names, or constructs. Moreover, it is unclear which principles apply to universal conceptual modeling. For example, complexity management, cognitive integration, and dual coding, all appear to serve specific functions in a language, which may not result in the language being universal. A dedicated analysis focusing on the need to support everyone and every

modeling use case is required to develop specific principles tailored to universal conceptual modeling.

3 Principles of universal conceptual modeling

3.1 Method for developing principles of universal conceptual modeling

To develop UCM principles, we began by identifying relevant theoretical foundations. Prior research has considered the theoretical foundations of conceptual modeling to include philosophy, psychology, linguistics, logic, semantics, semiotics, and engineering (design) [37, 68, 72, 107, 124, 160, 165]. These foundations underscore the role of conceptual modeling in representation, communication, problem solving, and design [28, 65, 72, 107, 114, 127, 146, 154]. Therefore, we reviewed research in these disciplines with the aim to distill universal principles and to ensure they are consistent and align with prior research. Figure 1 presents the process of developing the principles of UCM.

Our approach was guided by the general notion of “universal.” According to Merriam and Webster [45], universal is defined as “including or covering all or a whole collectively or distributively without limit or exception,” “present or occurring everywhere,” or “comprehensively broad and versatile.” The Oxford English Dictionary [123] defines “universal” as “extending over or including the whole of something specified or implied, esp. the whole of a particular group or the whole world, comprehensive, complete; widely occurring or existing, prevalent overall.” We define *universal* as a proposition about an object asserting the presence of the object in every setting at any point in time or relevance of the object for every setting at any point in time. In addition to the concept of universal, we considered other relevant concepts (e.g., “general-purpose,” “inclusive,” “accessible”).

Notably, *universal* is an aspirational claim about an object: Virtually nothing is absolutely universal (perhaps with the exception of physical energy and fields). This, however, suffices for our goal to define the principles of universal conceptual modeling. These principles become objectives to be attained and the more a conceptual modeling language adheres to these principles, the more universal the language becomes.

Guided by our definitions, we assembled relevant search keywords that combined the latter keywords with those related to the theoretical foundations of conceptual modeling. Using Google Scholar, we searched for terms such as “universal language,” “general purpose language,” “universal cognition,” “universal sign system,” “universal forms,” “universal design,” as well as their reasonable combinations and related terms (e.g., “ontological primitives”). This was an

iterative process, because reviewing this literature was useful in identifying other relevant sources mentioned by these works (e.g., the theory of generative design [174]).

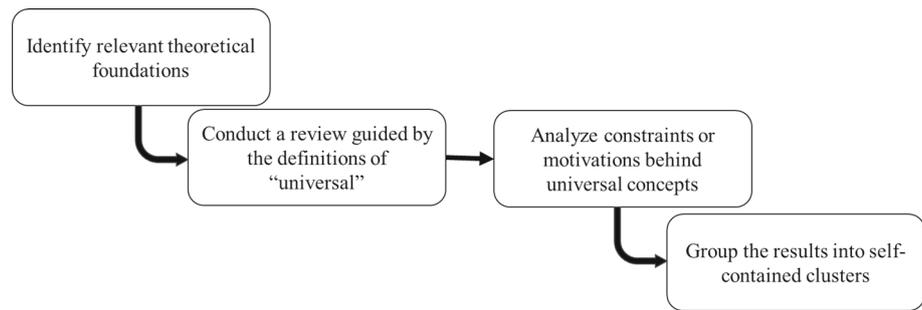
The results of this review are diverse and interdisciplinary—ranging from universal art forms and music to studies of mathematics, culture, design, and natural and artificial languages. Next, we reviewed the results to ascertain the potential of these works to constrain or motivate principles *behind* these universal concepts. At the end of this process, several themes emerged, originating mainly from the disciplines of anthropology, linguistics, semiotics, psychology, philosophy, and design. We draw primarily on theories from these disciplines.

In addition to the source of the principles, the review attested to the possibility of creating a universal data modeling language. Diverse interdisciplinary investigations suggest much universality and commonality among the elements that make up conceptual models or make them effective. Studies discuss “universal cognitive structures,” “cultural universals” (e.g., [151]), “cognitive universals” (e.g., [13]), “semantic universals” (e.g., [60]), universal “ontological primitives” (note, the concept of universals has a different meaning in ontology) (e.g., [82]), “universal design” [118], and a variety of taxonomies and ontologies of these elements (e.g., [43, 92]). These suggest that tapping into these ideas could offer design guidance for UCM.

To distill the principles from the literature, we followed the stopping rules approach established in requirements elicitation research [17]. We used “difference threshold” described as “[i]n gathering requirements for a new information system, a systems analyst interviews users until he gauges when he is no longer learning new information” [17, 91]. In this case, new information was understood as new principles for ensuring that conceptual modeling facilitates uses by anyone and for anything.

This process was supported by the “mental list” approach: “a mental list of items that must be satisfied before [the analyst] will stop collecting information” [17, 91]. The mental list was the list of theoretical foundations of conceptual modeling, all of which we wanted to consider. At this point, we were not restricted to any particular type of conceptual modeling (e.g., data, process). For the list of theoretical foundations of conceptual modeling, we consulted seminal publications in the conceptual modeling discipline that suggested theoretical foundations of conceptual modeling. Based on this literature, we obtained the following list: linguistics, philosophy, psychology and cognitive science, semiotics, engineering, and logic [28, 37, 72, 105, 110, 114, 124, 160, 165]. As we extracted the relevant principles from the literature, we wanted to ensure the theoretical foundations of conceptual modeling were represented in these principles.

Fig. 1 Process of developing the principles of UCM



The candidate principles identified through this process were then grouped into six clusters, or dimensions, of universal modeling and were subsequently labeled. From this, we propose principles of universal conceptual modeling (UCM).

3.2 Principles of universal conceptual modeling

We now provide the foundations for universal conceptual modeling in the form of six universal conceptual modeling principles: accessibility, flexibility, ubiquity, minimalism, primitivism, and modularity.

3.2.1 Accessibility

Most research reviewed converged on the notion that any universal form of communication, sign system, set of ideas, needs to be accessible, or easily understood, appreciated, and used. Empirically, the relationship between the accessibility features of ease of learning, ease of application, and widespread use has been established in education, linguistics, anthropology, psychology, as well as information systems studies.

Research shows that universal conceptual systems can be described using highly accessible and intuitive language and using accessible concepts. Anthropology, philosophy, and psychology, for example, studied universal concepts under the notion of *basic-level categories*. The research on basic-level categories suggests that there is a small list of “universal” concepts (e.g., “bird,” “tree”) [136], insofar as they are present in virtually every culture [11]. Furthermore, they are invariant to language proficiency and domain expertise: Both experts and novices understand the meaning of these categories and can use them effectively. They tend to be the first words patterns used when talking with children, and hence, the first basic words used by children [86]. Among other things, what makes the basic-level universal is its “accessibility”—its ease of understanding and use. The terms that express these concepts tend to be short, morphologically simple (e.g., car vs. sport utility vehicle). Furthermore, these terms map to what psychologists call “naturally occurring clusters” or “natural seams of the world” [112]: object referenced by these terms look alike and unlike the other “basic”

objects in the same environment (e.g., trees look very different from birds, but somewhat similar to each other). This makes it easy to apply these concepts without creating much communication confusion. Notably, owing to these properties, basic-level categories have already been considered in conceptual modeling research [29], with the aim to promote usability of models, especially for novices.

Linguistics, anthropology, and psychology, in particular, investigate “linguistic universals” and “semantic universals,” the latter being defined as the properties of meaning shared by the languages of the world [58, 152]. These are terms, such as every, some, most, the, a, three, and present in virtually all languages and all cultures. Steinert-Threlkeld and Szymanik [152] provide an explanation of the presence of such universals due to their ease of learning, showing that expressions satisfying universals are *simpler* than those that do not. Notably, information technology scholars have developed a model of technology acceptance predicated on the ease of use (with learning being a component) of the technology [42, 164].

Related to the notion of accessibility is the theory of “universal design” [118], which states that design needs to be maximally inclusive and sensitive to all potential users. This means designs should address the requirement of marginalized communities on the grounds that if the design works for them, it should also work for others [118]. There are indeed many groups that are not considered as typical designers and users of conceptual models, such as IT novices [75], people with cognitive limitations due to age or illness, and visually impaired people (most models are visual in nature) [143].

Considering the research on accessibility and the aspirations of modeling for everyone, we propose:

Principle 1 (Accessibility): Conceptual modeling (e.g., conceptual modeling languages, language constructs, methods, rules, symbols, tools, instructions) should be accessible (easy to understand, learn, and apply) to the broadest possible range of designers and users.

3.2.2 Flexibility

Despite many perceptible differences between human languages, there are properties shared by all languages, including common core at the level of syntax, semantics, and

phonology [7, 36, 76]. For example, despite a wide array of sounds across cultures, “universal phonetics” is present in all human societies, such as “all languages have stops” after the words and sentences [1, 76]. We already encountered one general explanation for this when dealing with basic-level categories: Commonality among humans is in part based on the mapping of the recurring patterns in the environment. Another mechanism behind these universals is the flexibility of form, as explained by the universal grammar theory.

A prominent group of linguists led by Noam Chomsky holds that underlying all natural languages have so-called *universal grammar* (UG)—innate principles that are instantiated when a speaker learns a particular language [35]. The UG theory describes a universal meta-language and defines the general principles and parameters (e.g., that every word can be identified with a linguistic category), which then become instantiated into specific languages. The UG theory suggests that a language aspiring to be general should permit *large variation* of expressions based on *few* principles and parameters. However, none of the principles and parameters are strictly binding, in that a given linguistic expression can deviate from them. Flexibility is desirable when it is impossible to predict exactly how a given language is going to be used by different people and in different situations. Among other things, flexible principles and parameters allow for great cultural diversity of human languages and permit for language evolution—all desirable characteristics for a language such as Danish.

Flexibility has also been suggested as a dimension of universality by other theories, most notably by theories of artifact design. For example, the theory of tailorable design suggests for tailorable technologies (flexible systems that can be modified by a user in the context of use) to be based on a dual design perspective (building some design functionalities, but allowing significant design choices to be defined by the user) [56]. Norman [118] promotes “universal design” and argues that “[t]he best solution to the problem of designing for everyone is flexibility: flexibility in the size of images and computer screens, in the sizes, heights and angle of tables and chairs. Allow people to adjust their own seats, tables and working devices.” In the same vein, Gregor and Jones [63] suggest a widely cited template for building design theories (theories of design and action). According to the authors, for design theories to be applicable to broader contexts, they need to be “malleable”—able to be modified, flexible.

Notably, flexibility is already considered in conceptual modeling research as a way to expand model uses and users. Hence, “flexibility” of conceptual models and the corresponding database schemas has been suggested as important to support highly heterogeneous user environments, such as social media, and crowdsourcing [97]. Similarly, flexibility in process modeling has been a design feature to support flexible organizational processes [39]. In the latter context,

flexibility was defined as: “the ability of the workflow process to execute on the basis of a loosely, or partially specified model, where the full specification of the model is made at runtime, and may be unique to each instance” [141]. Flexibility through setting parameters and general intentions, rather than specific instructions, has been suggesting as a generally applicable approach to IT development [145]. This strategy is consistent with agile and DevOps methodologies [9, 87].

Underlying the flexibility prescriptions in design theories is the assumption that it is virtually impossible to predict all possible ways to interact with an artifact.³ Hence, by promoting flexibility and malleability of a conceptual model, more uses and users can be accommodated. In this sense, the theories of linguistics and design converge: to ensure systems of communication, or signs are widely used, they need to be malleable and flexible.

Flexibility stands in opposition to the common “closed world” assumption underlying many computing languages (programming, conceptual), where rules of the language are explicitly established and are strictly binding [133]. In contrast, we encourage flexible and creative use, adaptation, and evolution of UCM. As with UG and universal design, such openness and flexibility are fundamentally important for ensuring that UCM can cover a broad range of uses and is accessible to the widest audiences. This background motivates the following general principle of UCM:

Principle 2 (Flexibility): Conceptual modeling should be based on few constraints, and local interpretation, adaptation, and evolution of conceptual modeling (languages, methods, tools) should be expected and encouraged.

3.2.3 Ubiquity

In addition to accessibility and flexibility, universality is understood in reference theoretical foundations as ubiquity—widespread, high frequency of occurrence and use and commonality of the signs, concepts, and ideas.

As mentioned, psychology, anthropology, and philosophy study basic-level categories. One reason for the acquisition and use of basic-level categories is their ubiquity, or high frequency of occurrence in the general discourse. These arguments originate in the work of Zipf [173], who foreshadowed the development of the basic-level theory by finding that more frequently used words tend to capture many of the terms needed for thinking and communication. Later, high usage frequency became a hallmark of basic-level categories [86]. Now, it is established that the relationship between frequency

³ According to theories of tailorable and secondary design, any artifact is a fundamentally multipurpose entity [56, 57]. Although an artifact may have an initial purpose, humans (and artificial entities) can always devise new ways to use the artifact, by rediscovering the new affordances of its features. For example, a wall once used to separate spaces could be turned into a museum exhibit.

and universality is two-way: Frequent words, which tend to describe basic-level categories, are more readily available for acquisition and reference, which enhances the universality of basic-level categories [112].

Converging arguments and evidence from numerous domains, including linguistics, psychology, and economics, suggest that cognitive and linguistic forms that are more frequently occurring tend to be the most important socially and economically. These are captured by Zipfian or power-law distributions [116, 169]. Summarizing these effects, the physicist Buchanan [18] proclaims that ubiquity of something reflects a simple, but highly successful generative mechanism. Things in social or physical domains would not be ubiquitous unless they follow some fundamental universal laws [18]. Although we may not always be able to explicate these mechanisms, ubiquity can serve as a proxy for these universal laws or patterns.

The ubiquity principle is widely found in history and anthropology. For example, there is a small set of symbols nearly universally shared across cultures. Specifically, “shapes found across many cultures include lines, circles, spirals, zigzags, squares and squares of circles” [135]. These shapes have a relatively stable and common general and contextualized understanding (e.g., a circle in the night sky is commonly assumed to be the moon). Furthermore, the manner in which the symbols are drawn exhibit strong regularity within (e.g., by different age groups) and across cultures. For example, parallel sides and right angles are considered basic compositional forms for graphical elements, known as the “geometric regularity effect,” a recent discovery in science [44, 139, 140].

The ubiquity principle suggests the constructs, visual notation, and rules used in a universal language should be ubiquitous; that is, drawn from those frequently used by different people, in different situations and settings.

Principle 3 (Ubiquity): Conceptual modeling should be ubiquitous, based on ideas, symbols, and concepts that are maximally familiar and widely used by all possible designers and users.

3.2.4 Minimalism

A corollary to the ubiquity principle is the existence of a core of elements, a small set of ubiquitous elements that account for much of communication and thinking. Another dimension of universality is minimalism—using the fewest possible constructs, signs, terms, visual symbols to communicate information.

The minimalism principle is a foundation of the *generative systems* theory. Generative systems (e.g., the number system, the Internet) have the capacity to produce both expected and unexpected variation as a result of open and fundamentally unpredictable use [157, 174]. According to the theory of

generative systems, the single most important principle of a generative system design is *minimalism* or “procrastination” [31, 174]. The idea is to only endow a generative system with essential features for functioning and to permit these features to evolve or be extended by its users. This principle has a strong rationale: the presence of too many choices may create high cognitive load and impede the adoption and use of a universal system, especially by novice designers.

Minimalism has additional support in the limited cognitive resources of humans. Psychology research suggests humans are capable of operating with and memorizing seven, plus or minus two, chunks of information [109]. Hence, seeking to limit the essential number of constructs to between five and seven (or less) can facilitate better learning and use of a universal language.

Principle 4 (Minimalism): Conceptual modeling should be based on few, essential elements, which by the ubiquity principle, should be the most widespread.

3.2.5 Primitivism

Conceptual modeling languages seek to represent concepts about the world (cf. [127]). To be universal, these concepts (and therefore the models that contain them) should be the most basic, primitive ones, describing basic facts, and structures of the world.

We define *primitivism* as a principle of conceptual modeling that seeks to capture those concepts that are most fundamental, basic, and atomic, in terms of which other concepts are defined. For example, the concept of a *business firm* is not primitive as it requires the prior understanding of its constitute notions, including an organization, people, resources, association, legal authority, and liability. In contrast, the concept of *resource* is more primitive which is also underscored by the fact that commonly a resource is defined by providing examples of the resources, such as money, materials, and other assets (e.g., [45]).

Whereas minimalism dictates having a small number of elements, primitivism suggests for the elements to be foundational, or the atomic and indivisible elements upon which all other elements are based. The more primitive the concepts, the more applicable they are in generic contexts, rather than specialized discourse. Hence, the concept of *business firm* has a restricted domain of modern economy, whereas the more primitive concept of *resource*, has a much broader domain, encompassing not only economy, but also such domains as production and biology.

Many theories dealing with universal forms are based on primitivism and thus provide the foundations and guidance on how to achieve primitivism in conceptual modeling. The “linguistic universals” and “semantic universals” suggest there are foundational elements in communication and thought [58, 152]. Hence, basic linguistic terms (*the, most*)

can be used to form more complex structures (e.g., *the most*). Geometric primitives, including *point* and *line segment*, are considered the foundational building blocks of visual forms [121].

Similarly, the cognitively universal basic-level categories are primitives in that they can be used to construct more complex cognitive structures [112]. For example, the basic (or primitive) concept of *tree* is the building block of a variety of more complex, or second-order concepts, such as *deciduous tree*, *blooming tree*, or *native tree*. Basic-level categories tend to be expressed with morphologically simple words, making it convenient to use them as a basic building block of verbal and written communication [136].

The idea of “ontological primitivism”—that there are basic building blocks of reality—can be found in philosophical theories about reality [10, 64, 68, 69, 102, 167]. Some philosophers believe the world can be reduced to its basic building blocks [21]. Others find an epistemological and methodologic justification: “every theory of nature needs to identify at least one ontological primitive, since we cannot keep on explaining one thing in terms of another forever” [82], p. 12.

Examples of ontological primitives include object, entity, or event. Different philosophers proposed lists of ontological primitives. Hence, Bertrand Russell [137] offered a complete “inventory of the world” consisting of “particulars, qualities, relations and facts.” For Ludwig Wittgenstein [171], a foundational element of reality was object. By far the most common philosophical approach is *substance ontology* [10, 48], which holds that the *basic element of being* is what is referred to as entity, object, thing, or substance. Some consider these terms synonymous. Halpin [70], for example, defines object as “any individual thing of interest.” For Bunge, object is the “undefined” primitive in terms of which other ontological constructs are defined. Conceptual modeling widely uses *object* as a key modeling construct. Object has been a core construct in conceptual modeling languages (e.g., UML, ORM, and OASIS) [70, 79, 91]. Object features as a key notion in a number of ontologies, such as UFO [65, 66], object-oriented ontology [71], and Basic Formal Ontology, BFO [149]. Hence, Guizzardi et al. [68] claimed “any reference theory for conceptual modeling would need a rich theory of entity (object) types.” Related concepts include entity, thing, system, these are present and are often considered primitive in a number of ontological theories (e.g., in BFO, UFO and Bunge), as well as foundational ideas of conceptual modeling [34, 70].

The concept is important in systems development more broadly (as in object-oriented analysis and object-oriented programming).

Principle 5 (Primitivism): Conceptual modeling should be based on fundamental semiotic, cognitive, visual, and ontological elements.

3.2.6 Modularity

The principle of primitivism in theories of universals is accompanied by the principle of modularity. For basic building blocks to be combined to form more complex, composite structures, they need to have modular design. Modularity is the property of having distinct self-contained components that can be combined together to create new forms.

Design research on extensible, generative technologies converge on an important principle of universal, flexible, and accessible systems—modularity of system components [56, 174]. Modularity allows systems to be combined and recombined, based on evolving and individual needs. It is also present in many natural systems: “the universal design strategy [of life] involves ‘bottom-up’ synthesis and modular, hierarchical organization both within and across multiple length-scales” [84]. For example, hierarchical organization of human concepts permits advanced thinking and reasoning and is the basis of DNA [134]. The ontological postulate that the world is made of systems explains reality as organized in modular levels, whereby basic elements of reality are layered on top of one another (e.g., chemical system is the basis for biological system) [22].

Due to its centrality in nature, modular design is a common basis of engineering [84]. Modularity permits the management of complexity, which can still be a challenge, even when the number of design choices are intentionally minimized. Hence, the modularity principle can be enhanced by flexibility—reducing constraints on how the components can be combined.

Modularity is further important for UCM design because of the constraints the other dimensions place. Specifically, primitivism, minimalism, and ubiquity suggest limiting the number of constructs. This makes it difficult to ensure important information is represented in models. It also promotes general use at the expense of specialization. However, modularity enables recombining the foundational building blocks, thereby creating the possibility for more complex rules and ideas to be captured. Hence, modularity is an important element of natural languages, whereby basic linguistic elements can be combined and recombined, in practically infinite ways [35]. Other universal systems are modular, such as mathematics, where basic axioms can be used to derive more complex constructs needed for specialized uses. Thus, a logarithm can be derived from more basic operations (e.g., multiplication, power). The result is the ability of modular systems to support a wide range of more specialized thoughts and expressions.

Principle 6 (Modularity): Conceptual modeling should be modular, supporting flexible and open combination of conceptual modeling elements (e.g., language constructs) in a variety of different ways.

4 Summary of the principles

The six principles—flexibility, accessibility, ubiquity, minimalism, primitivism, and modularity—constitute the theoretical basis of universal conceptual modeling. These principles are based on ideas from several disciplines and theories that have grappled with the notion of universality.

Each principle of UCM brings a unique consideration and is not reducible to another. For example, taking to the extreme, minimalism may suggest having only a single construct in the conceptual modeling language. At the same time, there could be multiple cognitive and ontological primitives, as a single ontological construct (e.g., object) alone cannot capture domain rules. Likewise, modular designs are only flexible if constructs can be combined in different ways. Similarly, ubiquitous approaches may not always be accessible, as they may systematically disenfranchise specific groups of users, while accessible and flexible approaches may be in the minority. Furthermore, the concepts that philosophy deems primitive might not be seen as accessible to broad audiences, such as the ontological concept of “perdurant.” Indeed, our version of Microsoft Word does not even include this word in its dictionary. Likewise, linguistic accessibility (e.g., specific concepts of “bird,” “tree”) may not correspond to constructs deemed ontological primitives (e.g., object, time, space, relationship, basic-level category in general) [48, 65, 136]. Taken separately, one or more of these principles does not guarantee UCM. To ensure conceptual modeling artifacts are as universal as possible, all the UCM principles need to be considered.

Although each UCM principle contributes uniquely to universal conceptual modeling, there is only partial independence among constructs. Instead, they converge by tapping into the shared essence of universality from different perspectives. For example, primitivism supports minimalism: There are fewer ontological primitives than derived constructs. Primitivism also facilitates modularity, as primitive constructs can be used to build more advanced structures, supporting specialized uses. If the process is guided by general parameters, as opposed to strict rules, it is consistent with the flexibility principle. Furthermore, accessible designs typically leverage flexibility. Likewise, modular forms, especially those found in nature, tend to be more ubiquitous. Considering this convergence, it is reasonable to hypothesize that the contribution of each principle toward the common core is cumulative. This means that a sound design strategy for the development of conceptual modeling artifacts is to adhere to all principles, thereby maximizing their benefits.

By distilling the principles of universal conceptual modeling from broad theoretical foundations, we are able to adopt converging interdisciplinary knowledge on universal systems. The principles permit us to advance the theory of conceptual modeling and support the design of better conceptual modeling artifacts. Next, we show the utility of these

principles. First, we use a case study to analyze the challenges of using Entity-Relationship diagrams and UML Class diagrams to represent open data and explore how following UCM principles can address these challenges. Second, the shortcomings of these languages with respect to the UCM principles motivate us to suggest modifications to existing conceptual modeling languages. We suggest how these principles can be used to guide the development of a new conceptual modeling language, *Datish*, built specifically with these principles in mind. Finally, we articulate additional directions for conceptual modeling research informed by the principles of universal conceptual modeling.

We note further that many of the principles explicitly consider the challenges faced by novice designers. For example, minimalism, accessibility, ubiquity, and modularity are directly informed by the needs of novice users. Modular designs (under the presence of the other principles) ensure that the building blocks of conceptual modeling can allow novices to begin using conceptual models, and more advanced features can be unlocked for specialized uses.

Table 2 summarizes the principles and explains how each contributes to either representation or language usage.

5 Evaluating principles of universal conceptual modeling

5.1 Evaluation strategy

We pursue a multi-faceted approach to evaluate the proposed principles. These principles are a type of abstract theoretical artifact called *meta-requirements* [77, 163]. Meta-requirements do not directly prescribe design choices, rather, they provide the theoretical justification and general foundation for design, analysis, and research in modeling. Consistent with this type of contribution, we show below how the principles can be fruitful in:

1. Evaluating the strengths and weaknesses of an existing conceptual modeling notation (shown in the case of RDF/OWL)
2. Guiding the design of a brand-new conceptual modeling language (shown in the case of *Datish*)
3. Identifying gaps in existing conceptual modeling literature and suggesting corresponding research opportunities (provided in the Research Directions section).

The applications of principles to advance the understanding of modeling practices, analysis and design of modeling languages, and identification of research opportunities provides evidence of the broad applicability and value of the proposed principles.

Table 2 Principles of UCM and their core benefit (highlighted) for conceptual modeling languages

Principle	Representation and semantics	Learning and usage
Flexibility	Allows language extensions and improvisations in language use	Allows diverse expressions and makes it more inclusive
Accessibility	Constrains representations to those which are accessible	Makes the language potentially usable to broadest audiences
Ubiquity	Constrains representations to those which are most common, frequently used	Makes the language potentially usable to broadest audiences
Minimalism	Focuses on very small number of constructs and rules	Makes the language easier to learn and use
Primitivism	Focuses on only basic, fundamental constructs	Makes the language more intuitive and general
Modularity	Allows the selection of modules based on representational needs	Makes the language easier to learn and use

5.2 Evaluating existing modeling languages: RDF

Equipped with the six principles of universal conceptual modeling, we can evaluate existing conceptual modeling languages and notations on their adherence to these principles. The principles of UCM provide a framework for systematically analyzing existing conceptual modeling notations. We illustrate this in the case of RDF (and identify an opportunity to conduct similar analysis for other conceptual modeling languages in the research directions below). RDF is a good example of a lean, expressive, and minimalist language which can be analyzed by the principles.

The Resource Description Framework (RDF) provides a basis for representing data so that it can be consistently accessed, interpreted, and shared.⁴ RDF statements are used for describing and exchanging metadata, which enables standardized exchange of data based on relationships. RDF is the primary foundation for the Semantic Web.

The core structure RDF is a set of triples—each consisting of a subject, a predicate, and an object—called an RDF graph. We now consider each of the six principles of UCM with respect to the RDF graph. Our goal here is to illustrate how the principles can guide the analysis of a language. Hence, we assume modeling with the RDF graph directly (as opposed to using this notation as a basis for another language).

Accessibility and Ubiquity. One of the challenges with RDF graph is accessibility and ubiquity. The framework was originally designed to support semantic interoperability on the web and presupposes a great deal of technical expertise. Even the most basic notions, such as “triplets,” “graph,” “node,” “predicate” are specialized concepts from relational algebra and mathematical set theory, which require some familiarity with these terms. Similarly, the definitions, such as of *subject* as “an IRI (Internationalized Resource Identifier)” or a “blank node” (“RDF 1.1 Concepts and Abstract Syntax” 2014), further assume familiarity with international resource identifiers and the architecture of the web in general.

Notably, the RDF graph can be made more accessible by providing more intuitive and non-expert friendly explanations and definitions, which can still map to the same underlying concepts. This, however, will only partially address ubiquity, as the basis for some of these concepts (e.g., URI) is rooted in specialized discourse.

Flexibility. The RDF graph is very flexible. It does permit modeling a variety of scenarios, especially for describing the structure of domains. However, the graph is especially conducive for the representation of static aspects of the domain and is limited for the representation of the change of structure over time.

Minimalism. The RDF graph is highly minimalist, containing only three elements. This is one of the leanest possible configurations, although not the absolute limit. It is feasible to imagine two constructs: object and description, as elaborated in the Danish language in the next section.

Primitivism and Modularity. The constructs of the RDF graph are known as ontological primitives, although not necessarily linguistic primitives (addressed in accessibility and ubiquity). However, one can question whether there is an overload in the meaning of these primitives. Specifically, a subject can also be an object. Thus, the RDF notation can be potentially simplified by merging and collapsing these constructs (if the aim is to optimize minimalism, as mentioned above). However, this separation of subject and object does provide a mechanism for capturing modularity. An object in one triplet can become a subject in another, thereby building conceptual relationships and structures.

In sum, our analysis of RDF using the proposed principles of universal conceptual modeling shows that, while flexible, minimalist, and modular, RDF graph presently struggles to be truly universal owing to its specialized foundation and aims. However, much potential exists in this language, and the analysis we conducted suggested several modifications to increase the universality of this language. This analysis

⁴ <https://www.w3.org/TR/PR-rdf-syntax/Overview.html>

demonstrated the value of the principles of UCM for uncovering the strengths and weaknesses of an existing notation from the novel perspective of universality.

5.3 Guiding the design of a new language: Datish

The six principles of UCM provide the foundation for the development of a new conceptual modeling language. A complete language design is beyond the scope of this paper because our focus is on UCM. Rather, this brief exposition illustrates how the UCM principles can be used to develop new conceptual modeling solutions. We leave the development of conceptual modeling languages following the design principles for future research.

An overarching implementation strategy is considering all principles together. Many converge; others may conflict. A design choice consistent with one principle can be strengthened or attenuated based on corroboration or conflict from the other principles. The aim is to introduce constructs that satisfy as many principles as possible. Broadly, for promoting representation or learning and usage, Datish should have a small number of constructs (following minimalism and primitivism to represent essential semantics), while striving for a flexible, accessible, and modular presentation to make it easier to learn and use.

Flexibility suggests that a language should support local interpretation, adaptation, and evolution. This principle can guide Datish design and be repeated at different points of the language description and instruction for learning. Flexibility also means that a local interpretation of this language is permissible and consistent with the design philosophy and, therefore, should be encouraged. It is a novel and counter-intuitive principle; therefore, it is important to clearly signal this unusual design feature to users.

Flexibility can factor in many ways in the Datish design. The constructs of the language can be used in any order, in any variation. For example, the language can permit representing only individuals, irrespective of what categories they belong, or only categories. The representations of individuals do not have to agree with the templates defined by the categories.

Accessibility involves, among other considerations, paying special attention to marginalized communities, including analysts with special needs (e.g., due to cognitive impairments), modeling novices and members of the public. Here, some guidance has already been developed by the conceptual modeling community. These include semiotic clarity and semantic transparency by Moody [110]. Another example is provided by Castellanos et al. [29] who synthesized psychology research on basic-level categories and provided actionable guidelines for selecting basic classes. These procedures can be employed in the choice of constructs for

Datish. For example, concepts such as *attribute* and *relationship* are among the basic terms that would be elements of a universal language.

Ubiquity suggests selecting graphic representations from the basic repertoire of shapes (i.e., lines, circles, squares). These shapes can be articulated based on geometric regularity (i.e., using parallel sides and right angles). Hence, for example, a square should have straight corners.

Minimalism can be implemented by choosing only constructs that are indispensable for capturing key domain semantics. Here, guidance can come from a review of broad practitioner and academic literature on data modeling. For example, many conceptual modeling projects use concepts such as entities, objects, attributes, or roles. In specific applications, such as data storage, constructs such as *aggregate* [19] or *relation* [38] are central. Such a review can identify the recurring patterns in the use of modeling constructs with the aim to distill the most essential ones. Yet, such guidance needs to be considered critically because conceptual modeling languages have been historically dominated by assumptions, such as representation by abstraction—meaning that conceptual modeling symbols represent classes of things in the world, rather than individual entities [99, 126, 150]. This approach to representation may be unhelpful in a universal language design. Other guidelines can also be used to resolve some of the choices among concepts (e.g., ontological primitivism, conceptual accessibility).

Note that philosophy does not have a monopoly over ontological primitives. Valuable addition guidance can come from psychology, as well as other disciplines, including art and design. This is especially valuable when considering all of the UCM principles together, because the acceptance by other areas of the ontological primitives can signal a good candidate for a universal construct (e.g., [71]).

Underlying all the design choices is the notion of *modular design*. The universal language needs to use identifiable and clearly separable components. These would permit language users to utilize different components based on their needs, as well as build upon the components and extend the language to more specific uses. For example, a specific component can deal with specialized constructs, such as system [103].

We now illustrate how these principles can shape specific features of Datish. In real language construction, the design decisions need to be more carefully explained and rigorously evaluated.

Based on the principles of modularity, minimalism, and primitivism, Datish can have *Formative* and *Structural Modules*. The Formative Module shapes the form of the language and is comprised of two constructs—*object* and *description*. The Structural Module provides for more specialized uses and includes the constructs of *individuals*, *categories*, and *relationships*.

The core construct of Datish is object. This choice was the outcome of a broad search across diverse interdisciplinary literature and simultaneous consideration of different UCM principles. Consider ubiquity. By far the most common philosophical approach is the ontology of substance [48], which holds that the *basic element of being* is what is referred to as object. Object is also accessible in that most people have an intuitive understanding of this concept. An object can be defined as *anything can exist, be thought about, talked about, or acted upon* [23].

To be useful, an object in a model needs to carry some information. This is needed for the user of the model to understand what object in the world is being represented and from what perspective. An object, therefore, needs to be expressed with some additional information. We call this a *description*, an elaboration of an object that communicates some of its relevant aspect(s) by linguistic or extra-linguistic means.

Taken together, object and description constitute the *minimal* elements of Datish grammar. This is the simplest possible architecture for a language, and a strong adherence to the minimalism principle: any grammar being a conceptual system must have at least two logically connected components [25]. It is conceivable to create Datish models consisting only of objects along with their descriptions. Such minimalism may be particularly useful in cases where very little is known about the nature of these objects, or when showing their relationships may not be necessary. Object-description pair is also flexible, permitting to (crudely) describe virtually all conceivable objects in the world.

We can now leverage the modularity principle and create a hierarchy of objects. Hence, objects as modeling constructs can be individual, representing any specific, singular objects (e.g., mental, physical, or social) in the world. Objects can also represent collections or categories of singular objects. Categories organize individuals into clusters, thereby allowing these individual objects to be treated in a similar manner. All objects can be connected to one another via relationships. The constructs of individuals, collections, and relationships are also accessible and intuitive for non-experts. In the name of flexibility, singular objects and categories can be related in any way, with singular objects being independent of any categories, and categories being sometimes devoid of individuals (for cases such as social categories that precede the existence of their members). Figure 2 depicts a meta-model of Datish, represented as a UML class diagram.

Having only five constructs in the modeling toolkit offers a great deal of flexibility. At the same time, these constructs enable the modeling of a wide variety of scenarios, domains, and systems. The descriptions attached to all objects can be simple or complex, depending on the purpose of the modeling. The description need not be next to the object and can

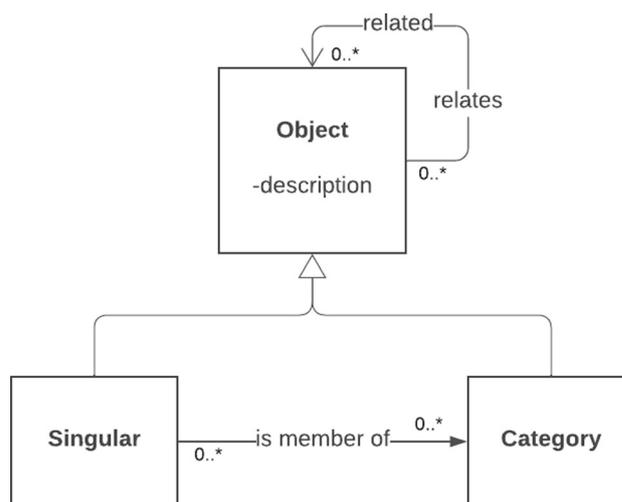


Fig. 2 Meta-model of Datish with rules shown

be in a separate section of a diagram (as common in architectural blueprints) or another representation [78] to permit longer descriptions.

The specific ways to arrange or express these constructs may vary from project to project. It is feasible that some organizations' analysts may customize their own styles or introduce situational or systematic constraints upon the usage of these constructs [142]. For example, some projects may insist on always modeling objects independent of categories, other projects may stipulate that every object must have a unique identifier attached. The flexibility of Datish explicitly enables and encourages these local choices. Extensions to the core of Datish are also encouraged. For example, it is conceivable to imagine that expert communities may extend the description construct to convey cardinality.

This illustration shows how the principles of universal conceptual modeling can be used to guide the development of conceptual modeling artifacts by informing specific design choices in a grammar for conceptual modeling language. The procedure can be pursued in future studies to develop new conceptual modeling languages with the aim of making them more accessible to broad audiences.

6 Research opportunities following the principles of universal conceptual modeling

As IT development increasingly becomes ubiquitous and used for non-IT professionals, supporting these expanded uses and users is paramount. One of the values of explicitly articulating the principles of universal conceptual modeling is in opening up new ways of thinking about conceptual modeling research and practice. First, the introduction of the principles raises questions about the nature of conceptual modeling and the assumptions behind its practices. Second, the principles permit reflection upon existing conceptual modeling languages, tools, and methods. Third, the principles may guide the development of new conceptual modeling languages, tools, and methods, where existing solutions appear do not offer sufficient support. These reflections afforded by the principles underlie the research opportunities and directions suggested below.

7 Research opportunity 1: evaluating and rethinking the nature of conceptual modeling and the assumptions behind its practices

The principles of universal conceptual modeling pose interesting challenges to the established theoretical assumptions of conceptual modeling research. Prior conceptual modeling research rarely considered the value of supporting anyone with conceptual modeling solutions. A popular assumption of conceptual modeling was that it belongs in the realm of particular uses and specific purposes. This is best exemplified by domain-specific conceptual modeling languages, methods and tools [14, 47, 53, 106]. Naturally, this assumption holds for many applications of conceptual modeling. However, with the opening of design to IT non-experts, such as organizational employees and the general public, and the emergence of practices such as offshoring, agile development, and DevOps, relaxing some conceptual modeling assumptions is needed.

The proposed principles offer an opportunity to reconsider traditional conceptual modeling approaches and suggest there could be conceptual modeling universals. We provided a high-level characterization of these universals in terms of principles that underlie conceptual modeling broadly. A corresponding opportunity is to offer more fine-grained, design-specific guidance on how to operationalize and apply these principles.

To pave the way for these efforts, we suggested some operationalization of the UCM principles. For example, for minimalism, it would be useful to identify the construct thresholds. We referenced one study suggesting that the number of constructs in a language should not exceed nine [109].

These considerations are mainly driven by constraints on human short-term memory. Future research can examine this recommendation empirically. Auxiliary theories can also be applied to potentially refine the recommendation about the minimal number of constructs.

The need to operationalize the principles can lead to exciting research questions related to cognitive and ontological primitives. Ontology has been a persistent foundation of conceptual modeling, with much conceptual modeling research involving the application of existing ontology, such as that of Mario Bunge [28, 166], or developing of new ontologies, such as UFO and Dolce [55, 68, 81]. However, thus far these efforts have not been explicitly guided by the ideas of universality. For example, a common assumption in ontology development is the need for exhaustive domain coverage. This of course is reasonable, but an interesting question is how to balance this assumption with the principles of minimalism, ubiquity, primitivism, and accessibility. Another challenge is the need to formally distill the minimal number of ontological primitives.

Another research opportunity is to deepen the conceptualization of the principles themselves. Considering the substantial convergence among the individual principles that comprise UCM, an interesting theoretical question is how these principles are related. It is conceivable that the contribution of each principle toward the common core is in some situations multiplicative, not just additive. This hypothesis can be evaluated empirically, and if confirmed, would validate our running assumption on the importance of maximizing the adherence to all principles simultaneously for specific applications, rather than prioritizing some principles, or even focusing on a subset at the expense of others. The instantiation of all principles, as opposed to some, could bring additional benefits to the respective conceptual modeling artifacts.

One of the valuable outcomes of developing principles of universal conceptual modeling can be a contribution to the discussion on greater inclusivity of conceptual modeling practices. The research on inclusive conceptual modeling and conceptual modeling for impaired users is emerging [95, 100, 143, 144, 147]. Our work offers specific guidance for addressing the challenges of inclusivity under the concept of universal conceptual modeling.

An important research opportunity is to study existing development projects through the lens of universal conceptual modeling. The principles of universal conceptual modeling motivate new questions to consider when investigating these projects: Who has been systematically sidelined by the practices of conceptual modeling? What potential users and uses are presently not supported by conceptual modeling artifacts due to the limitations in their designs? What rules, assumptions, facts, beliefs that stakeholders wish

to communicate and propagate into the features of information systems are currently not captured by existing conceptual modeling languages and methods.

These are some questions motivated by the principles of universality proposed here. As they demonstrate, the systematic ignorance of the need to support broad audiences could have created a theoretical vacuum in conceptual modeling research whereby certain unstated assumptions pervaded scholarship, resulting in a host of interesting questions being ignored.

8 Research opportunity 2: analysis of existing conceptual modeling artifacts

The principles of universal conceptual modeling can become the basis for the analysis of existing conceptual modeling languages, methods, and tools. One option is to use the six principles as the dimensions of analysis, which become the measures for the universality of an existing conceptual modeling artifact. The evaluation can be carried out analytically, as shown in the example above with RDF, or empirically. Both approaches have been employed in conceptual modeling and research on UCM can consult the methodological guidance that this literature produced [26, 27, 73, 93, 125].

Analytical investigations following the principles can consider the extent to which different constructs within a conceptual modeling grammar adhere to the different principles. In this sense, the constructs could receive a score or an index of conformity to the principles. For example, cardinality could be low on the accessibility score, whereas entity could be high on that score. In this way, specific suggestions for the design of conceptual modeling languages can be made. Consider one such suggestion—constructs that are deemed less universal can be withheld from the initial exposure by non-expert users (e.g., through a modeling tool), and be progressively unlocked as users gain more proficiency and experience with the conceptual modeling language. This example also demonstrates the possibility of engineering conceptual modeling methods informed by these principles. Such methods could be used for training conceptual modeling users, whereby existing languages are being taught progressively, in accordance with their adherence to the universal principles.

Recently, there has been growing interest in languages that are more generic in scope, as well as those that better support wider audiences. Consistent with these developments are such approaches as data virtual machines [31] and domain modeling [12]. Our principles can facilitate these developments and be used to assess the universality of these approaches along the dimensions corresponding to these principles.

In the case of empirical evaluation, one possibility is to assign the degree of universality to existing conceptual modeling languages. Dependent variables of such analysis could be ease of use, particularly by non-experts in IT and traditionally under-representative communities, as well as the breadth of application, especially in areas currently underrepresented by conceptual modeling.

9 Research opportunity 3: development of new conceptual modeling artifacts

An important use of the principles is to support the development of new conceptual modeling artifacts, most notably conceptual modeling languages, methods, and tools inspired by or directly grounded in the universal conceptual modeling principles.

Throughout the paper, we envisaged a potential conceptual modeling language, *Datish*. At this stage, *Datish* is underspecified, an ideal type of a universal conceptual modeling language capable of modeling anything by anyone. One research opportunity is the development of such a conceptual modeling language. This could be a standalone language, with self-contained constructs and rules. It could also be a basic module of other conceptual modeling languages.

The universal principles of conceptual modeling add new considerations for conceptual modeling language design. Indeed, some principles may be in opposition to other factors deemed valuable in a conceptual modeling language. For example, minimalism comes in opposition to the scope and coverage of UCM-based languages. Conceptual modeling research has traditionally privileged language expressiveness, under the assumption that all relevant constructs from the real world should be present in the language; otherwise “construct deficit” occurs, in which case some relevant facts cannot be modeled [110, 167]. However, while these ideas are relevant for developing complete representations in the context of traditional information systems development (typically by professionals for well-defined modeling purposes, such as database design), they are detrimental for a universal language. It is simply impossible to exhaust all the possible kinds of real-world objects and events and predetermine the requisite constructs in a universal language. These features of universal conceptual modeling lead to novel questions about the implications of “construct deficit” in UCM-based languages.

There are important reasons for asking these questions. Consider one aspect of conceptual modeling: presentation modality. Virtually all existing conceptual modeling languages are visual in nature [110, 143]. Yet, visual is only one of the human modalities. The World Health Organization

reports that globally 2.2 billion (~27.5% of the global population) people have visual impairment.⁵ A universal conceptual modeling language should be especially sensitive to these considerations.

Another consideration is the use of language by younger and older audiences. These demographics have been systematically ignored in information systems development in general. Research also documents the difficulties, in particular, older users face when using information technologies [155, 156]. Universal conceptual modeling, as modeling for everyone, should promote usage by traditionally marginalized communities and ensure its features are designed with these users in mind. In particular, flexibility is an important design consideration, as it has been shown in a variety of contexts, to be the key property for making designs more inclusive [118]. A more robust dialogue within conceptual modeling community on the matters of inclusive conceptual modeling can contribute to inclusive design theory and practice more broadly.

To be more inclusive, UCM-grounded conceptual modeling should be medium-agnostic and efforts are needed to develop additional ways of representing languages (e.g., via sound, text). This is especially important for supporting users with impairments. One solution can simply be tools that vocalize the symbols in the diagram. Potentially specialized devices can be design, such as tablets capable of communicating and recording visual information more efficiently.

Another intriguing solution is to have conceptual models that are natively designed in the non-visual modality. One possibility is a *haptic tablet* with tactile properties, such that conceptual modeling symbols can be hand manipulated. Another complimentary solution is to include conceptual modeling-supported AI systems [15], p. 202,[94, 96, 104], capable of dynamic interaction, powered by the natural language processing capabilities of large language models, LLMs [30]. A basic question that would naturally arise from such possibilities is whether these would still be considered conceptual models? One argument could be to suggest these AI-based models are tailored to the context of IT development in a way that generic AI solutions are not and that they are developed with the support of conceptual modeling languages. These possibilities raise more fundamental questions about what the boundaries of conceptual modeling are.

Empirical research in conceptual modeling has demonstrated the detrimental impact of, for example, construct deficit on the perceptions of utility and ease of use of conceptual modeling languages [132]. Here, many interesting questions emerge. Would these perceptions extend to a universal conceptual modeling language, especially when used by novices? Can construct deficit be mitigated through an

architecture that is flexible and modular? Is it possible that all existing conceptual modeling languages, including those considered by the empirical evaluations of construct deficit, lacked sufficient flexibility and modularity? In other words, the key shortcoming of these languages was not the construct deficit, but the inability to combine and recombine its basic building blocks to ensure that construct deficit does not occur? Finally, an interesting theoretical question is: Does a language that is especially able to flexibly combine and recombine its basic building blocks suffer from any construct deficit at all?

By articulating the principles of universal conceptual modeling, we provided the theoretical basis for the design of Datish. An important empirical question that stems from the development of such language is the ability to test the thesis that theory grounded conceptual modeling languages exhibit superiority on some important outcomes. For example, Bostrom [16] argued for software engineering to be based on grounded theoretical justifications in order to ensure the artifacts are more reliable. Bunge [24] suggested that designed artifacts should implement scientifically grounded rules in order to make their behavior more predictable. With Datish an opportunity will open to test these assumptions.

A key promise of universal conceptual modeling is modeling that is more inclusive, that is, able to accommodate diverse audiences. This means that a language such as Datish should be more accessible and usable by novices than languages that are less universal, as defined by the six dimensions of universality here. Supported by research on operationalization of these principles, this prediction can be tested empirically. The outcome of this test can contribute to the theory of conceptual modeling, universal design, as well as to the discourse on scientifically grounded engineering and design.

We summarize the research opportunities informed by the principle of universal conceptual modeling in Table 3.

10 Discussion and conclusion

This research reviewed important limitations in current modeling practices, along with new modeling opportunities and applications, to identify a promising direction of universal conceptual modeling (UCM). To support this type of modeling with sound theory, we proposed design principles of UCM. The principles are derived from a literature review across multiple disciplines dealing with universal design, cognition, signs, and forms.

The valuable outcome of the formulation of the UCM principles is the ability to evaluate existing, engineer new conceptual modeling artifacts, such as languages. We illustrated the latter possibility with the outline for a data modeling language, Datish. This preliminary work

⁵ <https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment>

Table 3 Universal conceptual modeling as a foundation for research opportunities

Research Opportunity	Suggested question to be addressed
Opportunity 1: Nature of conceptual modeling	<p>Reconsider how to perform conceptual modeling</p> <p>What are the unstated assumptions behind current conceptual modeling practices? Should they be revisited?</p> <p>What are the existing principles (minimalism, ubiquity, primitivism, accessibility) and should they be revisited?</p> <p>What is the role of ontology? Will it remain a consistent foundation?</p> <p>What will the role be of universal practices?</p>
Opportunity 2: Analysis of existing conceptual modeling artifacts	<p>Establish inventory of current approaches</p> <p>Evaluate current approaches to conceptual modeling</p> <p>What principles can be used to analyze existing conceptual modeling languages?</p> <p>How do we clarify constructs so that multiple constructs are not used in the same way?</p> <p>How can universal constructs be used in training?</p> <p>Is it possible to create a scale that would indicate the degree to which conceptual modeling languages are universal?</p>
Opportunity 3: Development of new conceptual modeling languages	<p>How can the principles be used to support the development of new conceptual modeling artifacts? What should those artifacts look like?</p> <p>Is it possible to construct a language that will accommodate users from many diverse demographics?</p> <p>How do we perform theoretical assessment of the strengths and weaknesses of a new language? Is it possible to put mechanisms in place that might overcome some of these challenges? What is the role of user feedback during the assessment process?</p> <p>How can UCM principles be updated as needed?</p> <p>Would perceptions extend to a universal conceptual modeling language, when used by novices?</p> <p>Can construct deficit be mitigated through an architecture that is flexible and modular?</p> <p>Do existing conceptual modeling languages, including those considered by the empirical evaluations of construct deficit, lack sufficient flexibility and modularity? How can the basic building blocks be combined?</p> <p>Does a language that is especially able to flexibly combine and recombine its basic building blocks suffer from construct deficit?</p>

demonstrates the potential for applying these principles in the development of a new language.

The principles can also be useful for the assessment of existing conceptual modeling languages. Some existing conceptual modeling languages, while not universal by our definition, are widely used. These include the popular ER model and UML. The principles of UCM can support strategies for teaching these languages and guide their extensions. For example, the notion of geometric regularity (a specific way to implement ubiquity) can be applied to the constructs of these languages to assess the general accessibility of these shapes. We anticipate that shapes inconsistent with the geometric regularity principle would be more difficult to learn, especially for novices. Similarly, many conceptual modeling languages already use ubiquitous shapes, but without theoretical justification. Moody [110], for example, laments that even such important and standard language as “UML does not provide design rationale for any of its graphical conventions [such as using a rectangle for the class construct].” Thus, the UCM

principles can be used to post-hoc rationalize and better evaluate the choices made “intuitively” by language designers in the past. (After all, geometric regularity and symbol universality effects were expected to influence these decisions).

Finally, the UCM principles contribute to the ongoing debate on the greater inclusivity of conceptual modeling practices [95, 100, 143, 144, 147]. Our work offers specific guidance for addressing the challenges of inclusivity under the concept of universal conceptual modeling and encourages continued refinement of conceptual modeling languages, methods, and tools, to making modeling more inclusive.

Appendix: Definition of key terms used in the paper

Table 4 in the appendix provides the definitions of key terms used in the paper. Note these definitions are subject to considerable debates, including within conceptual modeling community (e.g., [127]). Furthermore, different definitions of these terms exist in different disciplines and even within

Table 4 Definitions of key terms used in the paper

Term	Our Definition	References for our definition
Universal	A property of an object indicating the <i>presence</i> of the object in every setting at any point in time or <i>relevance</i> of the object for every setting at any point in time	Adapted from [45, 123]
Sign	A physical object that stands for another object (abstract or concrete)	From [119]
Grammar	Rules of combination and acceptable usage of the <i>signs</i> in a language	Adapted from [35, 117, 119]
Language	System of signs governed by <i>grammar</i> to communicate meaning and facilitate thinking	Adapted from [35, 119]
Concept	Mental abstractions representing things, classes of things, properties of things, states, and events, and other concepts	Adapted from [23, 112]
Conceptual modeling	Formal or semi-formal activity during information systems development and use that involves the usage of concepts to describe relevant aspects of reality for the purposes of understanding, communication, design, and decision making	Synthesis of [113, 131, 154]
Conceptual model syn: script	Description of relevant aspects of reality resulting from the execution of the conceptual modeling activity	Adapted from [168] and based on definition of <i>conceptual modeling</i>
Conceptual modeling symbol syn: construct	An artificial sign (commonly, non-iconic), used to designate <i>concepts</i> in conceptual modeling	Adapted from [20, 35, 117, 119]
Conceptual modeling language syn: notation	System of <i>conceptual modeling symbols</i> governed by <i>conceptual modeling grammar</i>	Adapted from <i>language</i> above
Conceptual modeling grammar	A formal system of rules that show how to use conceptual modeling symbols to produce conceptual models	Adapted from <i>grammar</i> above (cf. [168])
Conceptual meta-model	A type of conceptual model that abstracts the common properties of specific kinds of conceptual models (i.e., models generated using a particular grammar)	Adapted from [162]
Universal conceptual modeling	Conceptual modeling that can be carried out in any scenario by anyone	Adapted from <i>conceptual modeling</i> and <i>universal</i>

the same discipline. Hence, linguistics and semiotics define language as a formal system of signs but also as a tool of communication, see [117, 119]. In defining the terms here, we sought to consider the different perspectives on these terms, while whenever possible choosing the most widely accepted definitions. At the same time, our objective was to ensure the definitions are consistent with one another and with our intended usage of these terms. Hence, we provide our own definitions of these terms.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your

intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Akbar, M.A., Shad, M.K., Lai, F.W., Hussain, S.: Towards successful agile development process in software outsourcing environment: a systematic literature review. *Int. J. Bus. Innov. Res.* **23**(2), 141–167 (2020)
2. Al-Fedaghi, S.: In Pursuit of Unification of Conceptual Models: Sets as Machines. *ArXiv Preprint* <http://arxiv.org/abs/2306.13833> (2023)
3. Ambler, S.: *Agile Modeling: Effective Practices for Extreme Programming and the Unified Process*. John Wiley & Sons, Hoboken (2002)
4. Atzeni, P., Jensen, C.S., Orsi, G., Ram, S., Tanca, L., Torlone, R.: The relational model is dead, SQL is dead, and i don't feel so good myself. *ACM SIGMOD Rec.* **42**(1), 64–68 (2013)

5. Avison, D.E., Golder, P., Shah, H.: Towards an SSM toolkit: rich picture diagramming. *Eur. J. Inf. Syst.* **1**(6), 397–408 (1992)
6. Azevedo, C.L., Iacob, M.-E., Almeida, J.P.A., van Sinderen, M., Pires, L.F., Guizzardi, G.: Modeling resources and capabilities in enterprise architecture: a well-founded ontology-based proposal for ArchiMate. *Inf. Syst.* **54**, 235–262 (2015)
7. Barwise, J., Cooper, R.: Generalized Quantifiers and Natural Language. In: Kulas, J., Fetzer, J.H., Rankin, T.L. (eds.) *Philosophy, Language, and Artificial Intelligence. Studies in Cognitive Systems*, vol 2. Springer, Dordrecht (1981)
8. Beard, J. W., Storey, V. C., Samuel, B., Lukyanenko, R., Wiedemann, A., Schuff, D., Gunsey, S., Islamzada, Z., and Islamzada, F.: Agile development: the promise, the reality, the opportunity. In: Almeida, J., Claudio D., C., Kalloniatis, C. (eds.) *Advanced Information Systems Engineering Workshops CAiSE 2024 International Workshops*, Limassol, Cyprus, June 3–7, (2024)
9. Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., and Jeffries, R.: Manifesto for agile software development. In: *Agile Alliance*, Snowbird, UT. ([https:// agilemanifesto.org](https://agilemanifesto.org)) (2001)
10. Benovsky, J.: Relational and substantial ontologies, and the nature and the role of primitives in ontological theories. *Erkenntnis* **73**, 101–121 (2010)
11. Berlin, B., Breedlove, D.E., Raven, P.H.: Folk taxonomies and biological classification. *Science* **154**(3746), 273–275 (1966)
12. Børner, D.: *Domain Science and Engineering: A Foundation for Software Development*. Springer Nature, Berlin (2012)
13. Blaut, J.M., Stea, D., Spencer, C., Blades, M.: Mapping as a cultural and cognitive universal. *Annals Assoc. Am. Geogr.* **93**(1), 165–185 (2003)
14. Bork, D.: Metamodel-based analysis of domain-specific conceptual modeling methods. In: presented at the The Practice of Enterprise Modeling: 11th IFIP WG 8.1. Working Conference, PoEM 2018, Vienna, Austria, October 31–November 2, 2018, Proceedings 11, pp. 172–187. Springer (2018)
15. Bork, D.: Conceptual modeling and artificial intelligence: challenges and opportunities for enterprise engineering. In: *Enterprise Engineering Working Conference*, pp. 3–9. Springer (2022)
16. Bostrom, N.: *Superintelligence: Paths Dangers, Strategies*. Oxford University Press, Oxford (2014)
17. Browne, G., Pitts, M.G., Wetherbe, J.C.: Cognitive stopping rules for terminating information search in online tasks. *MIS Quarterly* **31**(1), 89–104 (2007)
18. Buchanan, M.: *Ubiquity: Why Catastrophes Happen*. Crown, New York (2002)
19. Bugiotti, F., Cabibbo, L., Atzeni, P., Torlone, R.: Database design for NoSQL systems. In: presented at the International Conference on Conceptual Modeling, pp. 223–231. Springer (2014)
20. Bunge, M.A.: *Treatise on Basic Philosophy: Semantics I: Sense and Reference*. Springer, Amsterdam (1974)
21. Bunge, M.A.: *Treatise on Basic Philosophy: Ontology I: The Furniture of the World*. Reidel, Boston (1977)
22. Bunge, M.A.: *Treatise on Basic Philosophy: Ontology II: A World of Systems*. Reidel Publishing Company, Boston (1979)
23. Bunge, M.A.: *Philosophical Dictionary*. Prometheus Books, Amherst (2003)
24. Bunge, M.A.: *Philosophy of Science: From Explanation to Justification*, vol. 2. Routledge, New York (2017)
25. Bunge, M.A.: Systems everywhere. In: Negoita, C. V. (eds.) *Cybernetics and Applied Systems* (pp. 23–41). CRC Press (2018)
26. Burton-Jones, A., Green, P., Parsons, J., Siau, K. (2017) Special issue on ontological analysis in conceptual modeling. Part 2. *J. Database Management* **28**(1): i–x
27. Burton-Jones, A., Wand, Y., Weber, R.: Guidelines for empirical evaluations of conceptual modeling grammars. *J. Assoc. Inf. Syst.* **10**(6), 495–532 (2009)
28. Burton-Jones, A., Weber, R.: Building Conceptual Modeling on the Foundation of Ontology. In: *Computing Handbook: Information Systems and Information Technology* (CRC Press.), Boca Raton, FL, United States, pp. 15.1–15.24. (2014)
29. Castellanos, A., Tremblay, M., Lukyanenko, R., Samuel, B.M.: Basic classes in conceptual modeling: theory and practical guidelines. *J. Assoc. Inf. Syst.* **21**(4), 1001–1044 (2020)
30. Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y.: A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, **15**(3), 1–45 (2024)
31. Chatziantoniou, D., Kantere, V.: Data virtual machines: data-driven conceptual modeling of big data infrastructures. In: presented at the EDBT/ICDT Workshops (2020)
32. Chatziantoniou, D., Kantere, V.: Data virtual machines: enabling data virtualization. In: Rezig, E.K., et al. (eds.) *Heterogeneous Data Management, Polystores, and Analytics for Healthcare. DMAH Poly 2021*. Springer, Cham (2021)
33. Chen, P.: The Entity-relationship model—toward a unified view of data. *ACM Trans. Database Syst.* **1**(1), 9–36 (1976)
34. Chen, P.: Entity-relationship modeling: historical events, future trends, and lessons learned. In: Broy, M., Denert, E. (eds.) *Software Pioneers*. Springer, Berlin (2002)
35. Chomsky, N.: *Knowledge of Language: Its Nature, Origin, and Use*. Greenwood Publishing Group, Connecticut (1986)
36. Chomsky, N.: *Syntactic Structures*. Walter de Gruyter, Basel (2002)
37. Clarke, R., Burton-Jones, A., Weber, R.: On the ontological quality and logical quality of conceptual-modeling grammars: the need for a dual perspective. *Inf. Syst. Res.* **27**(2), 365–382 (2016)
38. Codd, E.F.: A relational model of data for large shared data banks. *Commun. ACM* **13**(6), 377–387 (1970)
39. Combi, C., Gozzi, M., Posenato, R., Pozzi, G.: Conceptual Modeling of Flexible Temporal Workflows. *ACM Trans. Auton. Adaptive Syst. (TAAS)* **7**(2), 1–29 (2012)
40. Compagnucci, I., Corradini, F., Fornari, F., and Re, B. Trends on the usage of BPMN 2.0 from publicly available repositories In: presented at the International Conference on Business Informatics Research, pp. 84–99. Springer, (2021)
41. Davies, I., Green, P., Rosemann, M., Indulska, M., Gallo, S.: How do practitioners use conceptual modeling in practice? *Data Knowl. Eng.* **58**(3), 358–380 (2006)
42. Davis, F.D.: Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Q.* **13**(3), 319–340 (1989)
43. De Carlo, G., Langer, P., and Bork, D.: Rethinking model representation—a taxonomy of advanced information visualization in conceptual modeling. In: presented at the International Conference on Conceptual Modeling, pp. 35–51. Springer (2022)
44. Dehaene, S., Roumi, F.A.I., Lakretz, Y., Planton, S., Sablé-Meyer, M.: Symbols and mental programs: a hypothesis about human singularity. *Trends Cognit. Sci.* **26**(9), 751–766 (2022). <https://doi.org/10.1016/j.tics.2022.06.010>
45. Dictionary, M.-W.: *Merriam and Webster*, Springfield. Merriam-Webster, Massachusetts (2002)
46. Döbing, B., Parsons, J.: How UML is used. *Commun. ACM* **49**(5), 109–113 (2006)
47. Döller, V., Karagiannis, D., Utz, W.: MetaMorph: formalization of domain-specific conceptual modeling methods—an evaluative case study, juxtaposition and empirical assessment. *Softw. Syst. Model.* **22**(1), 75–110 (2023)
48. Dupré, J.: A process ontology for biology. *Philos. Mag.* **67**, 81–88 (2014)
49. Elahi, H., Wang, G., Chen, J.: Pleasure or pain? an evaluation of the costs and utilities of bloatware applications in android smartphones. *J. Netw. Comput. Appl.* **157**, 102578 (2020)

50. Eriksson, O., Johannesson, P., Bergholtz, M.: The case for classes and instances—a response to representing instances: the case for reengineering conceptual modelling grammars. *Eur. J. Inf. Syst.* **28**(6), 681–693 (2019)
51. Fettke, P.: How Conceptual Modeling Is Used. *Commun. Assoc. Inf. Syst.* **25**(1), 43 (2009)
52. Fettke, P., Reising, W.: Systems mining with heraklit: the next step. In: *BPM 2022 Forum*, Münster, pp. 89–104. Springer, Germany (2022)
53. Frank, U.: Domain-specific modeling languages: requirements analysis and design guidelines. In: Reinhartz-Berger, I., Sturm, A., Clark, T., Cohen, S., Bettin, J. (eds.) *Domain Engineering*. Springer, Berlin, Heidelberg (2013)
54. Frank, U.: Multi-perspective enterprise modeling: foundational concepts, prospects and future research challenges. *Softw. Syst. Model.* **13**, 941–962 (2014)
55. Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., Schneider, L.: Sweetening ontologies with DOLCE. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*. EKAW 2002. Springer, Berlin, Heidelberg (2002)
56. Germonprez, M., Hovorka, D.S., Collopy, F.: A theory of tailorable technology design. *J. Assoc. Inf. Syst.* **8**(6), 351–367 (2007)
57. Germonprez, M., Hovorka, D. S., and Gal, U.: *Theorizing On Secondary Design* (2009)
58. Gibson, E., Futrell, R., Piantadosi, S.P., Dautriche, I., Mahowald, K., Bergen, L., Levy, R.: How efficiency shapes human language. *Trends Cognit. Sci.* **23**(5), 389–407 (2019)
59. Giebler, C., Gröger, C., Hoos, E., Schwarz, H., and Mitschang, B.: Modeling data lakes with data vault: practical experiences, assessment, and lessons learned. In: *International Conference on Conceptual Modeling*, pp. 63–77. Springer (2019)
60. Goddard, C.: *Semantic Theory and Semantic Universals. Semantic and Lexical Universals*, pp. 7–29 (1994)
61. Gonzalez-Perez, C.: How ontologies can help in software engineering. In: Cunha, J., Fernandes, J., Lämmel, R., Saraiva, J., Zaytsev, V. (eds.) *Grand Timely Topics in Software Engineering. GTTSE 2015*. Springer, Cham (2017)
62. Greefhorst, D., Proper, E.: *Architecture Principles—The Cornerstones of Enterprise Architecture*. Springer, Berlin (2011)
63. Gregor, S., Jones, D.: The anatomy of design theory. *J. Assoc. Inf. Syst.* **8**(5), 312–335 (2007)
64. Guarino, N., Guizzardi, G.: In the defense of ontological foundations for conceptual modeling. *Scand. J. Inf. Syst.* **18**(1), 115–126 (2006)
65. Guizzardi, G.: *Ontological Foundations for Structural Conceptual Models*, Enschede, The Netherlands: Telematics Instituut Fundamental Research Series (2005)
66. Guizzardi, G.: Ontological meta-properties of derived object types. In: *Presented at the International Conference on Advanced Information Systems Engineering*, pp. 318–333. Springer (2012)
67. Guizzardi, G., Fonseca, C. M., Benevides, A. B., Almeida, J. P. A., Porello, D., Sales, T. P.: Endurant Types in Ontology-Driven Conceptual Modeling: Towards OntoUML 2.0. In: *Presented at the International conference on conceptual modeling*, pp. 136–150. Springer 2018
68. Guizzardi, G., Wagner, G., Almeida, J.P.A., Guizzardi, R.S.: Towards ontological foundations for conceptual modeling: the unified foundational ontology (UFO) story. *Appl. Ontol.* **10**(3–4), 259–271 (2015)
69. Guizzardi, R., Amaral, G., Guizzardi, G., Mylopoulos, J.: An ontology-based approach to engineering ethicality requirements. *Softw. Syst. Model.* **22**(6), 1897–2192 (2023)
70. Halpin, T.: *Object-Role Modeling Fundamentals: A Practical Guide to Data Modeling with ORM*. Technics Publications, Sedona (2015)
71. Harman, G.: *Object-Oriented Ontology: A New Theory of Everything*. Penguin UK, London England (2018)
72. Henderson-Sellers, B.: Why philosophize; why not just model?. In: Johannesson, P., Lee, M., Liddle, S., Opdahl, A., Pastor López, Ó. (eds.) *Conceptual Model. ER* (2015)
73. Hevner, A., Parsons, J., Brendel, A.B., Lukyanenko, R., Tiefenbeck, V., Tremblay, M.C., vom Brocke, J.: Transparency in design science research. *Decis. Support. Syst.* **182**(1), 1–19 (2024)
74. Hirschheim, R.: “Offshore outsourcing: challenge to the information systems discipline”, information systems outsourcing: enduring themes, new perspectives and global. *Challenges* **2**, 687–699 (2006)
75. Hvalshagen, M., Lukyanenko, R., Samuel, B.M.: Empowering users with narratives: examining the efficacy of narratives for understanding data-oriented conceptual models. *Inf. Syst. Res.* **34**(3), 890–909 (2023)
76. Hyman, L. M.: *Universals in Phonology*. UC Berkeley Phonology Lab Annual Report, Walter de Gruyter GmbH & Co. KG (2008)
77. Iivari, J.: Distinguishing and contrasting two strategies for design science research. *Eur. J. Info. Syst.* **24**(1), 107–115 (2015)
78. Jabbari, M., Recker, J., Green, P., Werder, K.: How do individuals understand multiple conceptual modeling scripts? *J. Assoc. Inf. Syst.* **23**(4), 1037–1070 (2022)
79. Jacobson, I., Booch, G., Rumbaugh, J.: *The Unified Software Development Process*, vol. 1. Addison-Wesley, Reading MA (1999)
80. Johnson, J.: *CHAOS 2020: Beyond Infinity* Centerville, MA: The Standish Group. <https://www.standishgroup.com/news/49> (2020)
81. Jonsson, T., Enquist, H.: Phenomenological ontology guided conceptual modeling for enterprise information systems. In: *presented at the International Conference on Conceptual Modeling*, pp. 31–34. Springer (2018)
82. Kastrup, B.: An ontological solution to the mind-body problem. *Philosophies* **2**(2), 10 (2017)
83. Kaur, K., Rani, R.: Modeling and querying data in NoSQL databases. In: *Big Data, 2013 IEEE International Conference On*, pp. 1–7. IEEE (2013)
84. Kushner, A.M., Guan, Z.: Modular design in natural and biomimetic soft materials. *Angew. Chem. Int. Ed. Chem. Int. Ed.* **50**(39), 9026–9057 (2011)
85. Lankhorst, M.M., Proper, H.A., Jonkers, H.: The anatomy of the archimate language. *Int. J. Inf. Syst. Model. Des. (IJISMD)* **1**(1), 1–32 (2010)
86. Lassaline, M. E., Wisniewski, E. J., and Medin, D. L. 1992. “Basic Levels in Artificial and Natural Categories: Are All Basic Levels Created Equal?”. In: Barbara, B. (ed.) *Percepts, Concepts and Categories: The Representation and Processing of Information* (Vol. Volume 93), *Advances in Psychology*, North-Holland, pp. 328–378
87. Leite, L., Rocha, C., Kon, F., Milojicic, D., Meirelles, P.: “A Survey of DevOps Concepts and Challenges. *ACM Comput. Surv. (CSUR)* **52**(6), 1–35 (2019)
88. Leondes, C.T.: *Manufacturing and Automation Systems: Techniques and Technologies*. Academic Press, Seattle WA (1991)
89. Li, Y., Currim, F., Ram, S.: Data completeness and complex semantics in conceptual modeling: the need for a disaggregation construct. *ACM J. Data Inf. Qual.* **14**(4), 1–21 (2022)
90. Lima, L., Miyazawa, A., Cavalcanti, A., Cornélio, M., Iyoda, J., Sampaio, A., Lewis, V.: An integrated semantics for reasoning about sysml design models using refinement. *Softw. Syst. Model.* **16**, 875–902 (2017)

91. Lopez, O. P., Hayes, F., and Bear, S.: Oasis: an object-oriented specification language. In: International Conference on Advanced Information Systems Engineering, pp. 348–363. Springer (1992)
92. Lorenzatti, A., Abel, M., Fiorini, S. R., Bernardes, A. K., and dos Santos Scherer, C. M.: Ontological primitives for visual knowledge. In: presented at the Brazilian symposium on artificial intelligence, pp. 1–10. Springer 2010
93. Lukyanenko, R., Parsons, J., and Samuel, B. M.: Artifact sampling in experimental conceptual modeling research. In: International Conference on Conceptual Modeling, pp. 199–205. Springer (2018)
94. Lukyanenko, R., Samuel, B. M., Storey, V. C., Sturm, A.: Conceptual modeling systems: a vision for the future of conceptual modeling. In: ER'2022 Forum, pp. 1–5. Hyderabad, Virtual (2022)
95. Lukyanenko, R., Bork, D., Storey, V.C., Parsons, J., Pastor, O.: Inclusive Conceptual Modeling: Diversity, Equity, Involvement, and Belonging in Conceptual Modeling. In: *ER Forum 2023*, pp. 1–4. Springer, Lisbon Portugal (2023)
96. Lukyanenko, R., Castellanos, A., Storey, V. C., Castillo, A., Tremblay, M. C., Parsons, J.: Superimposition: augmenting machine learning outputs with conceptual models for explainable AI,” In: 1st International Workshop on Conceptual Modeling Meets Artificial Intelligence and Data-Driven Decision Making, pp. 1–12. Springer, Vienna (2020)
97. Lukyanenko, R., Parsons, J.: Is traditional conceptual modeling becoming obsolete? In: *Conceptual Modeling*, pp. 1–14. Springer, Berlin (2013)
98. Lukyanenko, R., Parsons, J.: Beyond micro-tasks: research opportunities in observational crowdsourcing. *J. Database Management (JDM)* **29**(1), 1–22 (2018)
99. Lukyanenko, R., Parsons, J., Samuel, B.M.: Representing instances: the case for reengineering conceptual modeling grammars. *Eur. J. Inf. Syst.* **28**(1), 68–90 (2019)
100. Lukyanenko, R., Parsons, J., Storey, V.C., Samuel, B.M., Pastor, O.: Principles of Universal Conceptual Modeling. In: *EMMSAD 2023*, Saragosa, pp. 1–15. Springer, Spain (2023)
101. Lukyanenko, R., Parsons, J., Wiersma, Y.F., Wachinger, G., Huber, B., Meldt, R.: Representing crowd knowledge: guidelines for conceptual modeling of user-generated content. *J. Assoc. Inf. Syst.* **18**(4), 297–339 (2017)
102. Lukyanenko, R., Storey, V.C., Pastor, O.: Foundations of information technology based on Bunge’s systemist philosophy of reality. *Softw. Syst. Model.* **20**(1), 921–938 (2021)
103. Lukyanenko, R., Storey, V.C., Pastor, O.: System: a core conceptual modeling construct for capturing complexity. *Data Knowl. Eng.* **141**, 1–29 (2022)
104. Maass, W., Storey, V.C.: Pairing conceptual modeling with machine learning. *Data Knowl. Eng.* **134**, 101909 (2021). <https://doi.org/10.1016/j.datak.2021.101909>
105. Mandelburger, M.M., Mendling, J.: Cognitive diagram understanding and task performance in systems analysis and design. *MIS Q.* **45**(4), 2101–2157 (2021)
106. Mannadiar, R., Vangheluwe, H.: *Domain-specific engineering of domain-specific languages*. In: presented at the Proceedings of the 10th Workshop on Domain-Specific Modeling, pp. 1–6 (2010)
107. Mayr, H. C., Thalheim, B.: The triptych of conceptual modeling. In: *Software and Systems Modeling*, pp. 1–18. Springer (2020)
108. Michael, J., Bork, D., Wimmer, M., Mayr, H.C.: Quo vadis modeling?: Findings of a community survey, an ad-hoc bibliometric analysis, and expert interviews on data, process, and software modeling. *Softw. Syst. Model.* **23**(1), 7–28 (2023). <https://doi.org/10.1007/s10270-023-01128-y>
109. Miller, G.: The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychol. Rev.* **63**, 81–97 (1956)
110. Moody, D.L.: The ‘physics’ of notations: toward a scientific basis for constructing visual notations in software engineering. *Softw. Eng. IEEE Trans. On* **35**(6), 756–779 (2009)
111. Muehlen, M. zur, Recker, J.: How much language is enough? Theoretical and practical use of the business process modeling notation. In: Bubenko, J., Krogstie, J., Pastor, O., Pernici, B., Roland, C., Sølvsberg, A. (eds) *Seminal Contributions to Information Systems Engineering*, Springer, Berlin, Heidelberg (2013)
112. Murphy, G.: *The Big Book of Concepts*. MIT Press, Cambridge, MA (2004)
113. Mylopoulos, J.: Conceptual modeling and telos. In: Loucopoulos, P., Zicari, R. (eds.) *Conceptual Modeling, Databases, and CASE: An Integrated View of Information Systems Development*, pp. 49–68. John Wiley & Sons Inc, New York, NY (1992)
114. Mylopoulos, J.: *Information Modeling in the Time of the Revolution*. *Inf. Syst.* **23**(3–4), 127–155 (1998)
115. Nalchigar, S., and Yu, E.: Conceptual modeling for business analytics: a framework and potential benefits. In: 2017 IEEE 19th Conference on Business Informatics (CBI), pp. 369–378. IEEE (2017)
116. Newman, M.E.J.: Power Laws, pareto distributions and Zipf’s Law. *Contemp. Phys.* **46**(5), 323–351 (2006)
117. Newmeyer, F.J.: Grammar is grammar and usage is usage. *Language* **79**(4), 682–707 (2003)
118. Norman, D.A.: *The Design of Everyday Things*. Bsic Books, New York, NY (2002)
119. Noth, W.: *Handbook of Semiotics*. Indiana University Press, Bloomington (1990)
120. Nunes, I., Moreira, A., Araujo, J.: Gire: gender-inclusive requirements engineering. *Data Knowl. Eng.* **143**, 102108 (2023)
121. Ohmer, M.M.: *Elementary Geometry for Teachers*. Sefton-Ash Publishing, London (1969)
122. Olivé, A.: *Conceptual Modeling of Information Systems*. Springer Science & Business Media, Berlin (2007)
123. *Oxford English Dictionary: Oxford English Dictionary*, Oxford, England: Oxford. <https://doi.org/10.1093/OED/3114863554> (2023)
124. Parsons, J.: An Information model based on classification theory. *Manage. Sci.* **42**(10), 1437–1453 (1996)
125. Parsons, J., Cole, L.: What do the pictures mean? Guidelines for experimental evaluation of representation fidelity in diagrammatic conceptual modeling techniques. *Data Knowl. Eng.* **55**(3), 327–342 (2005)
126. Parsons, J., Wand, Y.: Emancipating instances from the tyranny of classes in information modeling. *ACM Trans. Database Syst.* **25**(2), 228–268 (2000)
127. Partridge, C., Gonzalez-Perez, C., Henderson-Sellers, B.: Are conceptual models concept models?. In: presented at the International Conference on Conceptual Modeling, pp. 96–105. Springer (2013)
128. Pastor, O., Gómez, J., Insfrán, E., Pelechano, V.: The OO-method approach for information systems modeling: from object-oriented conceptual modeling to automated programming. *Inf. Syst.* **26**(7), 507–534 (2001)
129. Pastor, Ó., León, A.P., Reyes, J.F.R., García, A.S., Casamayor, J.C.R.: Using conceptual modeling to improve genome data management. *Brief. Bioinform.* **22**(1), 45–54 (2021)
130. Pastor, O., Molina, J.C.: *Model-Driven Architecture in Practice: A Software Production Environment Based on Conceptual Modeling*. Springer Science & Business Media, Berlin (2007)
131. Recker, J., Lukyanenko, R., Sabegh, M.A., Samuel, B.M., Castellanos, A.: From representation to mediation: a new agenda for conceptual modeling research in a digital world. *MIS Q.* **45**(1), 269–300 (2021)

132. Recker, J., Rosemann, M., Green, P., Indulska, M.: Do ontological deficiencies in modeling grammars matter? *MIS Q.* **35**(1), 57–79 (2011)
133. Reiter, R.: On closed world data bases, Ginsberg, M. L. (ed.) *Readings in Nonmonotonic Reasoning*, Morgan Kaufmann Publishers Inc, San Francisco, CA, USA. pp. 300–310. (<http://dl.acm.org/citation.cfm?id=42641.42663>) (1987)
134. Ridley, M.: *Genome: The Autobiography of a Species in 23 Chapters*. HarperCollins (2013)
135. Roberts, S.: Is Geometry a Language That Only Humans Know? *The New York Times*. (<https://www.nytimes.com/2022/03/22/science/geometry-math-brain-primates.html>) (2022)
136. Rosch, E., Mervis, C.B., Gray, W.D., Johnson, D.M., Boyesbraem, P.: Basic objects in natural categories. *Cogn. Psychol.* **8**(3), 382–439 (1976)
137. Russell, B. 2007. *Logic and Knowledge: Essays 1901–1950*, (R. C. Marsh, ed.), London, UK: Spokesman.
138. Sabegh, M.A.J., Recker, J.: Combined use of conceptual models in practice: an exploratory study. *J. Database Management (JDM)* **28**(2), 56–88 (2017)
139. Sablé-Meyer, M., Ellis, K., Tenenbaum, J., Dehaene, S.: A language of thought for the mental representation of geometric shapes. *Cognit. Psychol.* **139**, 101527 (2022)
140. Sablé-Meyer, M., Fagot, J., Caparos, S., van Kerkoerle, T., Amalric, M., and Dehaene, S.: Sensitivity to geometric shape regularity in humans and baboons: a putative signature of human singularity. In: *Proceedings of the National Academy of Sciences* (118:16), National Acad Sciences, p. e2023123118 (2021)
141. Sadiq, S., Sadiq, W., Orlowska, M.: Pockets of Flexibility in Workflow Specification. In: *International Conference on Conceptual Modeling*, Springer. pp. 513–526 (2001)
142. Samuel, B.M., Watkins, L., Ehle, A., Khatri, V.: Customizing the representation capabilities of process models: understanding the effects of perceived modeling impediments. *Softw. Eng. IEEE Trans. On* **41**(1), 19–39 (2015)
143. Sarioglu, A., Metin, H., Bork, D.: How inclusive is conceptual modeling? A systematic review of literature and tools for disability-aware conceptual modeling. In: *42nd International Conference on Conceptual Modeling (ER 2023)*, pp. 1–15 (2023)
144. Sarwar, S., Wilson, D.: systematic literature review on making and accessibility. In: *Proceedings of the 24th International ACM SIGACCESS Conference on Computers and Accessibility*, pp. 1–5 (2022)
145. Schwartz, M.: *War and Peace and IT: Business Leadership, Technology, and Success in the Digital Age. IT Revolution*, New York NY (2019)
146. Siau, K., Rossi, M.: Evaluation techniques for systems analysis and design modelling methods—a review and comparative analysis. *Inf. Syst. J.* **21**(3), 249–268 (2011)
147. Siu, A. F., Kim, S., Miele, J. A., and Follmer, S.: ShapeCAD: an accessible 3D modelling workflow for the blind and visually-impaired via 2.5 D shape displays. In: *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility*, pp. 342–354 (2019)
148. Slatin, J.M.: The art of ALT: toward a more accessible web. *Comput. Compos.* **18**(1), 73–81 (2001)
149. Smith, B.: Beyond concepts: ontology as reality representation. In: *International Conference on Formal Ontology in Information Systems*, pp. 73–84. Izmir, (2004)
150. Smith, J.M., Smith, D.C.P.: Database Abstractions: Aggregation and Generalization. *ACM Trans. Database Syst.* **2**(2), 105–133 (1977)
151. Stea, D., Blaut, J.M., Stephens, J.: Mapping as a cultural universal. In: Portugali, J. (eds.) *The Construction of Cognitive Maps*. Springer, Dordrecht (1996). https://doi.org/10.1007/978-0-585-33485-1_15
152. Steinert-Threlkeld, S., Szymanik, J.: Ease of learning explains semantic universals. *Cognition* **195**, 104076 (2020)
153. Storey, V.C.: Relational database design based on the entity-relationship model. *Data knowl. Eng.* **7**(1), 47–83 (1991)
154. Storey, V.C., Lukyanenko, R., Castellanos, A.: Conceptual modeling: topics, themes, and technology trends. *ACM Comput. Surv.* **55**(14s), 1–38 (2023)
155. Tams, S.: Helping older workers realize their full organizational potential: a moderated mediation model of age and it-enabled task performance. *MIS Q.* **46**(1), 1–30 (2022)
156. Tams, S., Grover, V., Thatcher, J.: Modern information technology in an old workforce: toward a strategic research agenda. *J. Strateg. Inf. Syst.* **23**(4), 284–304 (2014)
157. Teigland, R., Power, D.: *The Immersive Internet: Reflections on the Entangling of the Virtual with Society Politics and the Economy*. Palgrave Macmillan UK, New York NY (2013)
158. Teorey, T.J., Yang, D., Fry, J.P.: A logical design methodology for relational databases using the extended entity-relationship model. *ACM Comput. Surv.* **18**(2), 197–222 (1986)
159. Thalheim, B.: Towards a theory of conceptual modelling. *J. Univers. Comput. Sci.* **16**(20), 3102–3137 (2010)
160. Thalheim, B.: The theory of conceptual models, the theory of conceptual modelling and foundations of conceptual modelling. In: *Handbook of Conceptual Modeling: Theory, Practice, and Research Challenges*, Springer, pp. 543–577 (2011)
161. Van Der Zee, D.-J., Holkenborg, B., Robinson, S.: “Conceptual Modeling for Simulation-Based Serious Gaming”, *Decision Support Systems* (54:1), pp. 33–45. Elsevier, Amsterdam (2012)
162. Van Gigch, J.P.: *System Design Modeling and Metamodeling*. Springer Science & Business Media, Berlin/Heidelberg (1991)
163. Venable, J.: The role of theory and theorising in design science research. In: *DESRIST*, pp. 1–18. (2006)
164. Venkatesh, V., Morris, M.G., Davis, G.B., Davis, F.D.: User acceptance of information technology: toward a unified view. *MIS Q.* (2003). <https://doi.org/10.2307/30036540>
165. Wand, Y., Monarchi, D.E., Parsons, J., Woo, C.C.: Theoretical foundations for conceptual modelling in information systems development. *Decis. Support. Syst.* **15**(4), 285–304 (1995)
166. Wand, Y., Weber, R.: Mario Bunge’s ontology as a formal foundation for information systems concepts. In: Weingartner, P., Dorn, G. (eds.) *Studies on Mario Bunge’s Treatise*, pp. 123–150. Rodopi, New Haven (1990)
167. Wand, Y., Weber, R.: On the ontological expressiveness of information systems analysis and design grammars. *Inf. Syst. J.* **3**(4), 217–237 (1993)
168. Wand, Y., Weber, R.: Research commentary: information systems and conceptual modelling—a research agenda. *Inf. Syst. Res.* **13**(4), 363–376 (2002)
169. West, G.: *Scale: The Universal Laws of Life, Growth, and Death in Organisms, Cities, and Companies*. Penguin, London (2018)
170. Western, P.: Why the majority of data projects fail: the case for a universal data language. *Snowplow*. <https://snowplow.io/blog/project-failure-universal-data-language/>. (2022). Accessed 9 Jan 2023
171. Wittgenstein, L.: *Tractatus Logico-Philosophicus*. Harcourt, London, UK (1922)
172. Wüest, D., Seyff, N., Glinz, M.: FlexiSketch: a lightweight sketching and metamodeling approach for end-users. *Softw. Syst. Model.* **18**, 1513–1541 (2019)
173. Zipf, G.K.: *The Psycho-Biology of Language: An Introduction to Dynamic Philology*. Houghton Mifflin, Boston (1935)
174. Zittrain, J.: *The Future of the Internet—and How to Stop It*. Yale University Press, New Haven (2008)



Roman Lukyanenko is an associate professor at McIntire School of Commerce, University of Virginia. Roman investigates and develops theoretical and practical solutions in data management, artificial intelligence, and citizen science/crowdsourcing. Roman published over 140 conference and journal articles, including in *Nature*, *MIS Quarterly*, *Information Systems Research*, and *ACM Computing Surveys*. The work was recognized in international awards, such as Best Paper at

MIS Quarterly, Herbert Simon Design Science Award, AIS Best Dissertation of 2014, and INFORMS Design Science Award.



Binny M. Samuel is an associate Professor of Information Systems at the University of Cincinnati's Lindner College of Business. He earned a PhD degree from Indiana University's Kelley School of Business. He has worked in IT roles at Ford Motor Company and Indiana University. His research interests broadly entail: 1) the use of representations for systems development and analytics and 2) evaluating the design of software and systems used to accomplish work. He has published in *Communications of the ACM*, *IEEE Transactions on Software Engineering*, *Information Systems Research*, *Journal of the Association for Information Systems*, and *MIS Quarterly*, among others. <https://orcid.org/0000-0002-3223-4616>

published in *Communications of the ACM*, *IEEE Transactions on Software Engineering*, *Information Systems Research*, *Journal of the Association for Information Systems*, and *MIS Quarterly*, among others. <https://orcid.org/0000-0002-3223-4616>



Jeffrey Parsons is University Research Professor and Professor of Information Systems in the Faculty of Business Administration at Memorial University of Newfoundland in Canada. His research interests focus on how to better represent human conceptualizations of the world in data. His work on this and related topics has appeared in top journals in several disciplines, including *MISQ*, *ISR*, *Management Science*, *JAIS*, *ACM ToDS*, *IEEE TKDE*, and *Nature*. Jeff's

research has been recognized in several ways, including *MISQ* paper of the year, AIS Senior Scholars Paper Award, and the INFORMS Design Science Research Award. He is a Fellow of the Association for Information Systems, Distinguished Research Fellow from TU Dresden, Schoeller Senior Fellow, and an ER Fellow.



Veda C. Storey is a Distinguished University Professor, Tull Professor of Computer Information Systems, and Professor of Computer Science at 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. Dr. Storey is a member of AIS

College of Senior Scholars and a member of the steering committee of the International Conference on Conceptual Modelling and the Workshop on Information Technologies and Systems. She is a recipient of the Peter P. Chen Award, an ER Fellow, an AIS Fellow, and an INFORMS Fellow.



Oscar Pastor is Full Professor and Director of Internationalization and Transference of the Valencian Research Institute of Artificial Intelligence (VRAIN) at the UPV in Valencia (Spain). He received his PhD in 1992 and was a researcher at HP Labs, Bristol, UK. His research activities focus on conceptual modeling, web engineering, requirements engineering, information systems, and model-based software production. He is an internationally recognized researcher in the domain of

Conceptual Modeling, being ER Fellow since 2010 and having been awarded with the Peter C. Chen in 2017. He is currently leading a multidisciplinary project linking Information Systems and Bioinformatics, oriented to designing and implementing tools for Conceptual Modeling-based interpretation of the Human Genome information.



Araz Jabbari is an assistant Professor of Information Systems at the Faculty of Business Administration (FSA), Université Laval. His research focuses on the design and use of digital technologies, examining how individuals and organizations navigate digital transformation. He is interested in the designing and application of systems and the role of representations in facilitating communication and clarification. Araz's work has been published in journals,

including *MIS Quarterly*, the *Journal of the Association for Information Systems*, *Decision Support Systems*, *Communications of the Association for Information Systems*, and the *Journal of Database Management*. <https://orcid.org/0000-0002-2172-0600>