

## Representing instances: the case for reengineering conceptual modelling grammars

Roman Lukyanenko<sup>a</sup>, Jeffrey Parsons<sup>b</sup> and Binny M. Samuel<sup>c</sup>

<sup>a</sup>Department of Finance & Management Science, Edwards School of Business, University of Saskatchewan, Saskatoon, Canada; <sup>b</sup>Faculty of Business Administration, Memorial University of Newfoundland, St. John's, Newfoundland, Canada; <sup>c</sup>Lindner College of Business, University of Cincinnati, Cincinnati, OH, USA

### ABSTRACT

While many conceptual modelling grammars have been developed since the 1970s, they share the general assumption of *representation by abstraction*; that is, representing generalised knowledge about the similarities among phenomena in a domain (classes) rather than about domain objects (instances). This assumption largely ignores the fundamental role that instances play in the constitution of reality and in human psychology. In this paper, we argue there is a need for a grammar that explicitly recognises the primary role of instances. We examine the limitations of traditional class-based approaches to conceptual modelling, especially for modern information environments. We then explore theoretical and practical motivations for instance-based modelling, and show how such an approach can address the limitations of traditional modelling approaches. We conclude by calling for the engineering of instance-based grammars as an important direction for conceptual modelling research to address the limitations of traditional approaches, and articulate five challenges to overcome in such efforts.

### ARTICLE HISTORY

Received 4 January 2016  
Revised 13 April 2018  
Accepted 7 June 2018

### ACCEPTING EDITOR

Par Agerfalk

### ASSOCIATE EDITOR

Jan Marco Leimeister

### KEYWORDS

Conceptual modelling; information systems analysis and design; requirements elicitation; database design; information quality; ontology; cognition

## 1. Introduction

Conceptual modelling has traditionally played a central role in information systems (IS) development (Burton-Jones, Recker, Indulska, Green, & Weber, 2017; Checkland & Holwell, 1998; Clarke, Burton-Jones, & Weber, 2016; Eriksson & Agerfalk, 2010; Mumford & Henshall, 1979; Rossi & Siau, 2000; Wand & Weber, 2002). Over the years, many kinds of conceptual models have been developed—data models, process models, models of business activity and goals, and models of enterprise and systems architecture (see, e.g., Azevedo et al., 2015; Burton-Jones & Weber, 2014; Davies, Green, Rosemann, Indulska, & Gallo, 2006; Mylopoulos, Chung, & Yu, 1999; Pentland, Recker, & Kim, 2017; Recker, Rosemann, Green, & Indulska, 2011; Soffer, Kaner, & Wand, 2008; Taghavi & Woo, 2017). In this paper, we focus on a prominent kind of conceptual modelling—models of “substance and form” (sometimes also referred to as “conceptual data models”) (Burton-Jones & Weber, 2014; Peckham & Maryanski, 1988). These models, by representing entities in the world and their properties, are especially effective for communicating the structure and organisation of application domains.

Models of substance and form (henceforth simply referred to as conceptual models) are typically diagrammatic representations (also known as scripts<sup>1</sup>)

and are widely used in IS development, both to facilitate communication and improve domain understanding between developers and users, and to guide IS development activities such as database design, user interface design, and programming (Aguirre-Urreta & Marakas, 2008; Batra, 2008; Kung & Solvberg, 1986; Mylopoulos, 1998; Rizzi, 2007; Rossi & Siau, 2000; Wand & Weber, 2002).

Since the early 1970s, scores of conceptual modelling grammars—that is, formalisations of constructs and rules on how the constructs can be combined to create a model of a domain—have been proposed, including popular ones such as the Entity-Relationship (ER) model (Chen, 1976), the Unified Modeling Language (UML) (Jacobson, Booch, & Rumbaugh, 1999), and Object-Role Modeling (ORM) (Halpin, 2007). Conceptual modelling grammars are also commonly part of broader systems of representations, such as enterprise modelling languages (e.g., ArchiMate enterprise modelling) (Lankhorst, Proper, & Jonkers, 2010) and knowledge management and reasoning systems (e.g., Sowa, 2000).

Nearly all existing grammars (including all popular ones) manifest the principle of representation by abstraction, whereby domain knowledge is represented via stylised statements (e.g., classes, entity types, relationships) about specific phenomena (e.g., objects, instances) (Mylopoulos, 1998; Peckham & Maryanski, 1988; Smith & Smith, 1977).<sup>2</sup> Yet, while

abstraction is a vital cognitive process humans use to reason about the world, it is dependent on a more fundamental process—that of recognising the existence and structure of individual objects or instances. Although both processes are central to attention, perception, memory, cognition, and ultimately, higher order domain understanding and reasoning (Kahneman, 1992; Murphy, 2004), the potential value of explicitly representing instances has not received adequate attention in conceptual modelling grammars.

In light of changes to the ways in which data are collected and used by organisations, there is a pressing need to revisit whether the central focus on classes in conceptual modelling is adequate for modern conceptual modelling tasks. Specifically, organisations increasingly rely on data generated outside historic organisational boundaries and norms (e.g., user-generated content) as a basis for analysis and decision-making (Abbasi, Zhou, Deng, & Zhang, 2018; Brynjolfsson, Geva, & Reichman, 2016; Levina & Arriaga, 2014). In addition, a growing class of systems and applications are being created for the age of “big data” (Chen, Chiang, & Storey, 2012; Jagadish et al., 2014), which often results in data being used for purposes other than those for which it was collected. New distributed and highly scalable approaches to storage and retrieval (e.g., NoSQL databases) are transforming the database industry (Atzeni et al., 2013; Badia & Lemire, 2011; Stonebraker, 2010). Another major trend is the increasing use of data analytics to perform advanced data transformations and discover unanticipated patterns using machine learning and advanced visualisation techniques (Larsen & Becker, *Forthcoming*; Provost & Fawcett, 2013). Finally, the interactions of users with systems, including in traditional, corporate settings, has become more complex, resulting in a growing need to allow for more varied user input to be recorded and stored (Burton-Jones & Volkoff, 2017; Germonprez, Hovorka, & Collopy, 2007; Rai, 2016). Traditional class-based conceptual modelling grammars were not developed for these kinds of applications, leading to the question of whether they can be modified to better accommodate the evolving IS landscape or whether new grammars are needed.

The rest of this paper is organised as follows. We first consider how instances have been handled in conceptual modelling and related disciplines. Next, to better understand the potential value of instance-based representations, we examine the benefits and shortcomings of representation by abstraction (i.e., class-based), which underlies most prevailing grammars. We then consider arguments from philosophy and psychology (two common reference disciplines of conceptual modelling) that indicate the need to promote instance-based representations. These arguments are used to show how a focus on instances can address the limitations of class-

based grammars and better support emerging applications. We argue that the absence of such grammars has potential negative consequences for various issues in IS development and use. We conclude by articulating challenges that need to be resolved in the development of an instance-based grammar.

## 2. Instances in conceptual modelling and related disciplines

We start by detailing the key difference between traditional conceptual modelling grammars (which we term class-based) and requirements for instance-based conceptual modelling grammars. Then, we highlight the use of instances in some reference disciplines of conceptual modelling.

### 2.1. Class-based vs. instance-based conceptual models

Most existing grammars, and the vast majority of research on conceptual modelling to date, have been grounded in the assumption of inherent classification (Parsons & Wand, 2000; Ramesh, Parsons, & Browne, 1999). The assumption of inherent classification can have a strong assumption and dictate that “specific things in the domain of interest (entities, objects, etc.) can be referred to only as instances of classes (variously referred to as entity types, categories, kinds)” (Parsons & Wand, 2000, p. 229). A consequence of making this assumption is that “data that do not fall into a [class]... have either to be subverted to fall into one, or they cannot be handled in the data model” (Tsichritzis & Lochovsky, 1982, p. 8). Alternatively, a grammar may make a weaker assumption of inherent classification, whereby an instance may not be constrained to have only the attributes of a class (i.e., it may have unique attributes), but class remains a focal construct and the primary way of modelling domains. In such cases, although the option to use instances independent of classes may be available, modellers may be dissuaded from using it widely, especially given the long-lasting tradition and the established nature of class-based modelling: “the notion of universal [i.e., class] underlies the most basic and widespread constructs in conceptual modeling” (Santos, Almeida, & Guizzardi, 2013, p. 694). Thus, the ArchiMate® 3.0.1 Specification advises modellers of the preferred usage of classes over instances (The Open Group, 2017, pp. 64–65):

*“[T]he ArchiMate language in general focuses on the modeling of types, not instances, since this is the most relevant at the Enterprise Architecture level of description. Hence a business object typically models an object type (cf. a UML class) of which multiple instances may exist in operations. Only occasionally, business objects represent actual instances of*

*information produced and consumed by behavior elements such as business processes. This is in particular the case for singleton types; i.e., types that have only one instance.”*

We refer to grammars that make inherent classification for their constructs as **class-based conceptual modelling grammars** (we also note that the assumption also applies to other constructs when representations of specific, unique or particular phenomena have to conform to an a priori generalised form—for example, a specific relationship between two entities is an instance of a relationship type).

We contrast such grammars with “instance-based” ones. We define **instance-based conceptual modelling grammars** as grammars that do not require representations of individual phenomena to conform to a priori abstractions. Such grammars explicitly promote representational uniqueness, whereby each representation of the same instance may be different (i.e., expressed using different attributes and classes), including different representations by the same user at different times. For example, an instance-based modelling grammar may allow a modeller to represent a class *dog* (using generalised statements or common attributes), but also allow that a particular dog has additional attributes (not possessed by dogs in general). Instance-based models may also be completely free of explicit abstractions such as classes (e.g., a model of only specific individuals and their unique attributes), or may use abstractions (e.g., classes) as merely labels for instances (e.g., indicating breeds of dogs, without modelling “breed” as a class).

In contrast to class-based approaches, incorporating instances in conceptual modelling has received little attention. This is quite surprising considering that support for representing instances, objects, particulars or individuals has a long tradition in reference disciplines for conceptual modelling, including logic, computer science, and artificial intelligence.

## 2.2. Instances in reference disciplines of conceptual modelling

In logic, a major reference field of conceptual modelling (see Clarke et al., 2016), predicate calculus has mechanisms to reason and represent both universals and particulars (Feigl, Scriven, & Maxwell, 1958) (e.g., to represent a specific dog, Dakota, that has a tail (by virtue of being a dog), one could use a formalism such as  $\text{dog}(\text{Dakota}) \rightarrow \text{hasTail}(\text{Dakota})$ ). Likewise, modern extensional logic, a more recent branch of logic, is specifically focused on instances (Pap, 1978; Quine, 1977).

Logical formalisms that support reasoning about individuals are common in knowledge representations and artificial intelligence (e.g., directed graphs). Strong

support for individuals can be found in several programming languages, including such early ones as Prolog/Datalog (Patel-Schneider & Horrocks, 2007). Notably, in Datalog, individuals can be declared without a reference to a class (e.g., *writtenApaper*(Ågerfalk, Eriksson) describes the fact that two individuals, Ågerfalk and Eriksson, have written a paper together). Sowa (1979, 2000) applied graph theory to database design by showing how specific records in a relational database (i.e., instances) can be converted into a knowledge graph that enables automatic reasoning and visualisation. Directed graphs are representations of ordered pairs of elements, and are general enough to capture data about individuals, classes, attributes and relationships. For example, any ER diagram can be represented using (labelled) directed graphs (Frisendal, 2016).

Representation of individuals is also a prominent feature of Semantic Web technology, where representing information resources (e.g., people in the world or specific Internet resources) is part of the Resource Description Framework (RDF). In RDF, individual things (representing corresponding information resources) can be described using triples of subject-predicate-object (Heath & Bizer, 2011). More recently, reasoning about and representing individual things have been driven by semantic web *search*. This effort was pioneered by Google, Microsoft and Facebook’s search engines, powered by reasoning over knowledge graphs (Aggarwal, Bhatia, & Misra, 2016; Steiner, Verborgh, Troncy, Gabarro, & Van De Walle, 2012), and is now being adopted for corporate information search and retrieval (Gomez-Perez, Pan, Vetere, & Wu, 2017).

## 2.3. Instances in IS conceptual modelling research

Notwithstanding the attention given to instances in other disciplines, the importance of instances has not received adequate attention in conceptual modelling. One potential explanation is provided by Sowa (1980). Conceptual models are often used to design databases and databases originally handled large volumes of data of uniform structure. This is in contrast to, for example, artificial intelligence, where applications tend to work on data of highly variable structure.

Typically, requirements elicitation for IS begins with use cases, which are positioned around specific scenarios (Elmasri & Navathe, 2009; Jacobson et al., 1999). Yet, traditional modelling suggests analysing use cases and other requirements primarily to extract entity types (classes) (Hirschheim, Klein, & Lyytinen, 1995; Olivé, 2007; Parsons & Wand, 1997a). Narratives (e.g., user stories) may still be used to support graphical elements of the model (Burton-Jones & Meso, 2008; Hvalshagen & Samuel, 2014; Hvalshagen, Samuel, & Lukyanenko, 2017), but the models themselves seek to depict

abstractions and make generalisations over the domain (Mylopoulos, 1998; Olivé, 2007).

Among conceptual modelling grammars, many adhere to the strong inherent classification assumption. Notable examples include UML object diagrams (when UML is used for modelling application domains) (Jacobson et al., 1999; Pitone & Pitman, 2005), as well as ER Set Diagrams (Elmasri & Navathe, 2009). While one may use these notations to represent specific instances, *by design* they restrict attributes and classes of instances to those defined in the corresponding class-based schema, so that they are inherently instances of classes. Thus, in practice analysts may see these diagrams as merely supporting the ER and UML class diagrams—which are used as primary modelling grammars (Davies et al., 2006; Dobing & Parsons, 2006).

Some grammars provide a more direct support for instances alongside classes, but continue to emphasise the primacy of classes—that is, they make a weak inherent classification assumption. ArchiMate, an Enterprise Architecture modelling language by The Open Group, depicts data objects akin to the class in UML class diagrams (Azevedo et al., 2015; Lankhorst et al., 2010; Meertens et al., 2012). The language provides mechanisms for the representation of instances, but encourages types where several instances may exist. Further, when instances are represented the notation is overloaded as no notational difference between types and instances is made.<sup>3</sup> This also suggests that instances are seen as supporting the primary class-based abstraction level used for the language's broader purpose of Enterprise Architecture. Similarly, Sowa and Zachman (1992) provide a mechanism to represent instances in conceptual graphs to illustrate ER diagrams with specific instances. When used in the context of modelling, however, the objective of the graphs is to enable mapping from other languages (e.g., ER, class-based relational database calculus) to natural language (Sowa, 2000; Sowa & Zachman, 1992), and not to analyse, capture, and represent a domain independent of the class-based relational model or ER notations (the motivation of the current research).

Early work on instance-based modelling, undertaken by Parsons and Wand, was guided by an explicit aim to challenge the assumption of inherent classification (Parsons, 1996; Parsons & Wand, 2000). However, this work was focused on logical representations to produce an instance-based data model. Lukyanenko et al (Lukyanenko, Parsons, & Wiersma, 2014a, 2017) extended these ideas by formulating general principles for modelling instances and applied them to a real-world case in citizen science crowdsourcing. Their research used a flexible database schema, but the development proceeded *without creating a conceptual modelling diagram*—

thus no conceptual modelling artefact was developed. Lukyanenko et al. argue that an instance-based approach to IS development should emphasise individuals rather than classes, and the resulting IS should represent specific *instances* via attributes as perceived by different users at different moments in time—that is, adopt the representational uniqueness assumption for data collection and storage. A consequence of representational uniqueness is that capturing class-based abstractions a priori is unnecessary.

Having developed an instance-based IS (i.e., data collection, user-interfaces and retrieval systems that focus on instances), Lukyanenko et al. (Lukyanenko et al., 2014a; Ogunseye, Parsons, & Lukyanenko, 2017) admit that developing instance-based systems without the support from conceptual modelling creates a serious gap. The authors argue that practitioners may struggle because the focal phase of IS development that converts requirements into formal specifications to guide subsequent stages of development, is notably absent. They conclude by calling for the development of instance-based conceptual modelling grammars.

Recognising the value of instance-based representations, practitioners have begun to design new grammars, and extend existing ones, to enable representation of instances. These efforts have taken on two principal forms: (1) visualising existing data collected as instances; and (2) extending existing modelling grammars to depict instances independent of classes.

First, there is a growing practitioner community that employs directed graphs to capture and communicate business rules and requirements. These efforts are largely propelled by the growing popularity of NoSQL databases, especially key-value pair and graph databases (e.g., DynamoDB, Neo4j) (Angles & Gutierrez, 2008; Frisendal, 2016; Grolinger, Higashino, Tiwari, & Capretz, 2013). NoSQL vendors (e.g., Linkurious powered by Neo4j, see <https://linkurio.us>) increasingly provide innovative tools for data visualisation of graph and key-value pair databases, which practitioners reuse to capture or communicate business logic—effectively turning these visualisations into conceptual models (Frisendal, 2016; Kaur & Rani, 2013). The support for instances in such models is often incidental—for example, a model can reuse the node construct to show both a class of things (e.g., banks) as well as an instance (e.g., Vladimir Antonov—a specific banker, see Villedieu, 2015). However, practitioners recognise their versatility and recommend using these visualisations for conceptual modelling over traditional modelling grammars such as ER or UML (Frisendal, 2016).

Second, practitioners have begun to explicitly recognise “lack of instances” (Hills, 2016, p. 70) as a deficiency in modern conceptual modelling grammars. In a recent textbook on NoSQL modelling,

Hills (2016) recommends increased attention to instances in models (p. 71):

*If one wishes to connect one's data models to the real world, it can be very important to represent, not only real-world entity types, but also particular instances of those entities – particular entities – and relationships to data about those instances.*

To overcome the “lack of instances” in grammars, Hills (2016) proposed a new notation—COMN (Concept and Object Model Notation). COMN, while resembling ER and UML class diagrams, reserves a separate construct for instances shown using labelled hexagons (as opposed to rectangles used to depict classes). Hills believes this visual separation aids in comprehension as it emphasises the standalone, independent (of classes) nature of instances. Hills also suggests to use the instance construct (1) to illustrate values of classes, as done in UML object diagrams, but also (2) to represent unique, one-of-a-kind instances that are unlike other members of the class.

Despite recent advances, instance-based modelling remains largely overlooked both in conceptual modelling theory and in practice. Contrary to the emerging use of graphs for modelling, some researchers doubt “the usefulness of data models” (Kaur & Rani, 2013, p. 3). Many NoSQL projects do not use any formal specification (Kaur & Rani, 2013; Lukyanenko & Parsons, 2013a; Lukyanenko et al., 2017). Yet others propose to use traditional class-based models in support of NoSQL databases (Bugiotti, Cabibbo, Atzeni, & Torlone, 2014). To date, the conceptual modelling community remains largely unaware of instance-based models (Kaur & Rani, 2013; Schram & Anderson, 2012). Recent developments, such as COMN or graph conceptual models, are grassroots and occurring with little structure and virtually no interaction with the conceptual modelling research community. They are also more pragmatic as opposed to theoretic in their design, thus specific design choices are not justified or supported with logical arguments or empirical evidence. Moreover, there is no framework of use cases to support adoption of instance-based modelling. The choice to use, for example, graph models, appears to be often driven by the availability of tools provided by a given database vendor. Yet, the emergence of interest in instance-based representations among practitioners strongly points to the value and need of instance-based conceptual modelling.

Beyond the initial theoretical work on instance-based representations and the grassroots efforts by practitioners to represent instances, little is understood about the potential for instance-based representations in conceptual modelling—a shortcoming we address in this work. In the next section, we describe advantages

and limitations of using class-based conceptual models. We then examine instances in theories from philosophy and psychology, well established reference disciplines of conceptual modelling, to motivate the consideration of instance-based models.

### 3. Theoretical foundations: classes and instances

Distinguishing individuals (instances) from abstractions is fundamental to human existence. The nature of instances and classes is a core subject in psychology and philosophy. We use theoretical foundations in these reference disciplines to highlight both the advantages and challenges of class-based models. We then present the motivation for instance-based representations from these reference disciplines.

#### 3.1. Advantages of class-based models

Many ideas from psychology and philosophy have been incorporated into conceptual modelling grammars (for discussion, see Browne & Parsons, 2012; Eriksson & Agerfalk, 2010; Henderson-Sellers, 2015; Moody, 2009; Wand, Monarchi, Parsons, & Woo, 1995). We highlight some key advantages of class-based models noted by prior research (see Table 1).

#### 3.2. Limitations of class-based models

Despite the advantages of class-based models summarized in Table 1, there are important limitations of class-based models based. However, in this section we highlight limitations of class-based models based on the same reference disciplines that have established the advantages of class-based models—psychology and philosophy. In general, the known advantages of class-based conceptual modelling become disadvantages if the representations overemphasise the primacy of classes (i.e., if they assume inherent classification). We believe the importance of instances is overlooked in class-based models, and the limitations of class-based conceptual models (described below) may be remedied by instance-based models.

##### 3.2.1. Preserving individuality and domain complexity

Classes ignore individual differences among instances and focus on their similarity. This advantage of class-based models, however, makes it difficult to rely on classes for capturing unanticipated, unique and one-of-a-kind objects (Falkowski & Feret, 1990; Hills, 2016; Lukyanenko et al., 2014a), or to consider an object in its own right, independent of classes to which it might belong.<sup>4</sup> This may preclude models from capturing valuable exceptional data for which no regularities have been established.

- *Limitation 1.1: Class-based models are generally unable to represent uniqueness of instances.*

Another consequence of the fact that classes abstract away from the diversity of instances is that instances must exist prior to this abstraction process. As discussed earlier, this abstraction is necessarily purpose-driven. It always involves loss of irrelevant (to the purpose) data (e.g., properties of instances not fitting the classification schema selected). As there could be potentially infinite ways unique instances making up reality can be partitioned (Bowker & Star, 2000; Lakoff, 1987; Ritvo, 1997), this makes class-based models susceptible to two other potential limitations:

- *Limitation 1.2: The purpose(s) for which the class-based model is constructed must be established in advance.*
- *Limitation 1.3: The purpose(s) for the class-based model should be relatively stable over time.*

Naturally, conceptual models may be updated to reflect changing IS requirements, but this is often costly, considering that this update is typically accompanied by the corresponding changes to the database schema, programming code and user interface elements. Even after the change, however, the new schema remains susceptible to the same limitations, as these are rooted in the fundamental properties of classes.

Also, reaching consensus among modelling stakeholders (e.g., users who provide data, decision makers, developers) on which aspects to represent and which to ignore is critical to this mode of representation. Both the completeness and the accuracy of class-based representations are contingent on the understanding and quality of information requirements. A major practical and research challenge is effectively engaging subject matter experts to identify and record all relevant requirements as accurately and completely as possible (Appan & Browne, 2010, 2012; Browne & Ramesh, 2002; Mylopoulos et al., 1999; Taghavi & Woo, 2017). Despite years of research on requirements elicitation, eliciting a complete and accurate specification of a domain is a formidable task in practice, considering the complexity of organisational settings, frequent disagreements among business actors on definitions and business goals, organisational politics, time and budget constraints during systems analysis, and varying levels of requirements elicitation abilities of the analysts.

A practical consequence of L1.2 and L1.3 is that IS built with models that no longer reflect current goals and purposes may be unable to capture properties of instances that were not part of the original class

definition (e.g., Lee, 2003; Lee, Pipino, Funk, & Wang, 2006; Strong & Volkoff, 2010) or, in some cases, may be unable to store entire instances if no classes for these instances were originally anticipated (e.g., Cardamone et al., 2009; Lintott et al., 2009; Lukyanenko, Parsons, & Wiersma, 2014b).

### 3.2.2. Mitigating context-setting biases

Classes provide context and act as filters upon reality as they direct selective attention to those features learned as part of the original definition of the class (Colner & Rehder, 2009; Duncan, 1984; Lavie, Hirst, De Fockert, & Viding, 2004). This means that when the environment changes, humans may have a tendency to anchor to known and familiar classes (Gilovich, Griffin, & Kahneman, 2002; Kahn & Sarin, 1988), and may therefore ignore attributes of instances that do not conform or are not necessary for the original learned class definitions (Ogunseye et al., 2017). Finally, as classes are important linguistic mechanisms that enable humans to refer to the intended instance, if the classes no longer fit the environment this ability may be compromised, and classes may hinder, rather than aid, setting the appropriate context. This may result in a lost opportunity to learn something new or to adequately represent a new or emerging phenomenon, or may result in storage of the wrong instance.

- *Limitation 2.1: Class-based models act as anchors, which may bias users' perceptions of phenomena.*

### 3.2.3. Promoting open domain boundaries

As discussed earlier, class-based models permit analysts to completely represent the domain of interest. However, this may have negative consequences. Users working with class-based models may expect or assume that the classes captured in a conceptual model exhaust the domain. This may not be the case due to incomplete requirements elicitation or unanticipated environmental changes that introduce instances into the domain for which no classes have been provided.

A consequence of the completeness property of conceptual models is that they result in domain boundaries that become embedded in other objects and features of the IS. As a result, users working with the system may be unable to enter data on instances that fall beyond the boundaries of the domain. This may stifle employee creativity. In the case of volitional IS use (e.g., in crowd-sourcing projects), users may be dissuaded from joining and participating in the project if they feel that their interests (i.e., instances and classes that are important for them), fall outside the scope of the online project (e.g., Lukyanenko, Parsons, Wiersma, Sieber, &

**Table 1.** Known advantages of class-based conceptual models.

Advantages of class-based models	Explanation
Simplifies the complexity of the real world	Humans are a “classifying animal” whose “continued existence depends ... on the ability to recognize similarities and differences among objects and events” (Berlin, Breedlove, Raven, & Hammel, 2013, p. 25).
Provides context in communication	Access to human reality is impossible without invoking abstractions (Searle, 1983, 1995). Modelling using class-based abstraction is intuitive and natural when reasoning and communicating about domains. For example, when attempting to communicate about some phenomenon <i>x</i> , people typically refer to <i>x</i> using some class (e.g., <i>tree branch</i> ) (Lukyanenko et al., 2014a).
Completely represents domains	The concept of a “complete” specification has long been a cornerstone of conceptual modelling (Parnas, 1972), and persists to this day (Clarke et al., 2016). Also, “A conceptual schema is the definition of the general domain knowledge that the information system needs to perform its functions; therefore, the conceptual schema must include all the required knowledge” (Olivé, 2007).
Promotes inferences of unobserved attributes	An important property of classes is inferential utility (Parsons & Wand, 2008). These inferences are generally not represented explicitly, thus reducing storage requirements, model complexity (Batra, 2007; Siau, 2004) and decreasing cognitive load for people working with conceptual models.
Helps create social realities	Classes are fundamental to representing social domains (Bergholtz & Eriksson, 2015; Eriksson, Johannesson, & Bergholtz, 2018; March & Allen, 2015) created by humans using declarative speech acts (Mattessich, 2013; Searle, 1995). For example, people declare elections, form institutions, create corporations, and pass laws. In each case, these institutions may be created by proposing “templates” or “blueprints” prescribing essential properties and functions for the social objects. In these cases, social classes are created before any instance of the classes exist (Bergholtz & Eriksson, 2015; March & Allen, 2012, March & Allen, 2015; Searle, 1995). Effectively, one can turn to conceptual modelling, a specific type of human language and symbol system, as a tool to engineer social reality.

Maddah, 2016). We summarise this as two additional limitations:

- *Limitation 3.1: Class-based models set domain boundaries that might prevent representation of classes and instances that do not fit within the boundaries.*
- *Limitation 3.2: Class-based models set domain boundaries that may inhibit user engagement.*

It is worth noting that an instance-based grammar will be, in practice, unable to depict the entire domain. This means that, by using instance-based representations, analysts may choose not to identify domain boundaries explicitly, and instead, show typical members of the domain. Such representations facilitate the expansion of the scope of the project and, unlike the use of class-based representation, do not inhibit user engagement, creativity and participation (Kleek, Styke, Schraefel, & Karger, 2011).

### 3.2.4. Ensuring correct inferences

An important advantage of classes is their support for inferences on unobserved attributes. This means class-based models typically communicate more information than is explicit in a schema. For example, a conceptual model containing a class *Bird* will not list every single attribute of the bird, relying instead on inferences associated with the class. For example, attributes such as *has wings* and *has feathers* might be omitted to conserve space and reduce model complexity.

Likewise, due to inferences, classification may be inappropriate in situations where assigning a class to an instance may be premature. A famous example is the debate in science on the nature of dark matter. Recently, scientists have questioned whether the label “dark matter” is helpful for the debate as this classification label carries inferences related to the typical behaviour of matter (Kroupa, Pawlowski, & Milgrom, 2012; Mannheim, 2006). It is entirely possible that what we now refer to as “dark matter” or “dark energy” does not possess the properties of matter or energy, respectively, and insisting on these labels can prevent progress in uncovering the nature of these phenomena.

- *Limitation 4.1: Class-based models may offer distorted impression of instances if inferences drawn are inappropriate for all instances.*

### 3.2.5. Domain understanding

An important traditional role of conceptual modelling is supporting domain understanding for individuals involved in IS development. Building an IS requires different types of knowledge: 1) knowledge of IS; and (2) knowledge of the application domain. IS knowledge is the understanding of representations, methods, techniques, and tools that are utilised to build an IS, whereas application domain knowledge is the understanding of the real-world setting (e.g., context) for which an IS is built (Khatri, Vessey, Ramesh, Clay, & Park, 2006). IS roles (e.g., systems analysts) tend to have a higher degree of IS knowledge and other IS stakeholders (e.g., users participating in IS development) tend to have a higher degree of application domain knowledge (Browne & Ramesh, 2002; Samuel, Watkins, Ehle, & Khatri, 2015). Models created from conceptual modelling grammars seek to bridge these different knowledge

bases by providing a systematic way for IS roles, which are responsible for the construction of the IS, to discover and document knowledge about the application domain from users (Topi & Ramesh, 2002). IS roles share these models with users to review their discovery and documentation with respect to the application domain. Users are asked to validate that the model documents the application domain accurately, completely, without conflicts, and without redundancy (Dobing & Parsons, 2006; Shanks, Tansley, & Weber, 2003). In essence, one purpose of a model is to serve as a mechanism to facilitate shared understanding between IS roles and users.

As suggested earlier, the process of creating current class-based models involves generating abstractions, which deliberately ignores individual differences among instances and represents only properties shared by groups of instances. However, due to abstraction, relying on models as a mechanism to facilitate shared understanding might prove challenging, both between IS roles and users *and* among IS roles. Prior research has referred to this limitation as “the gulf of evaluation” (Norman, 2002)—the reader of a model having difficulty recreating the concepts the model intends to portray. Psychology has debated there is more than one way in which individuals cognitively process and reason about concepts (i.e., classes in conceptual modelling) than the traditional *classical theory*<sup>5</sup> assumed by many traditional class-based conceptual modelling grammars (see Lukyanenko and Samuel (2017) for a review). For example, the *exemplar theory* suggests that humans can represent and process concepts as a set of instances, and each instance in a concept can be different from other instances (e.g., “things to be taken out of a burning house”). Forcing all phenomena in a domain to fit one particular rule of class-based formation may impede understanding. Borrowing from psychology, conceptual modelling research has already begun to question the efficacy of representations that emphasise classes alone given that understanding entails also using instances (Samuel, 2012a; Samuel, Khatri, & Ramesh, 2018).

- *Limitation 5.1: Reliance on class-based models alone can impede understanding of models.*

Considering the theoretical arguments exposing the limitations of class-based conceptual modelling above, we next motivate the importance of instance-based representations from the same reference disciplines.

### 3.3. Motivation for instance-based representations from philosophy and psychology

We suggest instance-based representations as a potential remediation of the limitations of class-based ones. In this section, we provide a motivation

for instance-based representations in IS from psychology and philosophy.

Prior arguments in conceptual modelling research for instance-based representations (Eriksson & Agerfalk, 2010; Lukyanenko et al., 2014a; Parsons, 1996; Parsons & Wand, 2000; Soffer, Wand, & Kaner, 2015) frequently draw on Mario Bunge’s ontology (Bunge, 1977, 2003, 1979; Wand & Weber, 1990). According to Bunge, material things (e.g., specific planets, birds, trees, atoms)—instances—are the primary constituents of reality. People use classes to group things with common attributes (properties accessible to human perception and imagination). In Bunge, things are observer-independent ontological primitives.

Other philosophers (e.g., John Searle) assume the world is made of subatomic particles, fields, and forces (Searle, 1995). According to Searle, unlike instances in physical world, in the case of social reality, classes may be created before instances. However, once instances of the social classes are created (so called, social objects), they take on their own existence. In time, these instances therefore may not fully adhere to the original class-based template and may thus take on the class-independence of physical instances.

Instance (object) perception and object awareness appear to be fundamental to human cognition and a precursor to a more complex process of classification. Psychologists argue that, while people experience continuous sensory input (e.g., light falling on retina, sound waves), they eventually transform this into separate and distinct representations (Harnad, 1990). As early as William James (nineteenth century), psychologists argued that people do not see a continuum of “blooming, buzzing confusion” but “an orderly world of discrete objects” (Harnad, 1990). Instances are also believed to be units of attention: humans perceive sensory fields (e.g., visual space) to be made of discriminable objects and an undifferentiated perceptual background (Carey, 2009; Kahneman, 1992). These objects are then supplied via the brain’s neural network to higher order processing mechanisms, including those making judgments about similarity and differences and forming classes (Barsalou, 2014; Hahn, 2014; Larkey & Markman, 2005; Schwering, 2005).

Instances are also important for day-to-day, naïve reasoning about reality. In day-to-day life, the level that is naturally accessible to humans is that of “middle-sized” objects. Middle-sized objects are those that can be “picked out using unaided human sensory capacities” (Foster, 2011, p. 1), such as trees, animals, or rocks. These objects exist between quantum and macro cosmic levels. The importance of material objects makes them a powerful metaphor for reasoning and thinking about mental and social worlds—the

imaginary or social objects may not have the same ontological status of real things, but humans are comfortable acting as if such instances in the physical world (Harnad, 1990; Searle, 1983, 1995).

We conclude that both from the point of view of composition of the physical world and human cognition, instances are primary in that they serve as building blocks for classification. This creates a foundation for the benefits of instance-based representations. In the next section, we use these benefits to show how we can overcome the theoretical limitations of class-based models identified earlier.

#### 4. Understanding the benefits of instance-based conceptual modelling

To illustrate the potential benefits of instance-based conceptual modelling, we present several hypothetical conceptual modelling examples, and in some cases, we employ elements of a known conceptual modelling grammar with potential instance-based constructs. Although it is unclear whether an instance-based modelling grammar should reuse and extend existing grammars (e.g., use the class construct from an established grammar), for illustration purposes we adopted elements from UML (e.g., notation for classes, generalisation/specialisation relationship). We model instances as filled in circles (Samuel, 2012a; Samuel et al., 2018) and later call on future research to explore effective ways to represent instances in conceptual models.

##### 4.1. Case 1: representing instance individuality

While classes are specifically used by humans to abstract from the individuality of instances to the shared features of a collection of instances, in many real-world scenarios representing individual characteristics of instances might be required for some applications. Doing so using only classes is impossible because classes depict similarity among instances.

In contrast, expressing instance individuality is possible by incorporating an instance-based approach into traditional modelling. To illustrate, we build on a case suggested by Hills (2016) which shows limitations of class-based models when modelling the US judicial system. The US judicial system is hierarchical and consisting of federal and state courts. Courts of the higher level (e.g., federal courts) review appealed decisions made at the lower level (e.g., state-level). However, the United States Supreme Court stands out as a unique court that does not share many of the characteristics of other courts. Figure 1 shows the relationship between the Supreme Court and other courts using UML Class Diagram with an extended notation to explicitly show instances.

Figure 1 provides a simplified model of the US judicial system implemented in a hypothetical UML-like instance-based grammar. As seen from the figure, this diagram would not be possible using traditional conceptual modelling grammars, such as ER diagrams or UML without violating the syntax. The United States Supreme Court (shown as a filled-in circle) is an instance of the Court class, but is quite different from other courts. It has unique behaviour (e.g., while other courts rule on cases, the Supreme Court passes opinions on appeal decisions by lower courts) and shares very few attributes with typical courts (e.g., has US Congress confirmed justices instead of judges). To faithfully represent this instance (with all its unique properties and relationships) and properly distinguish it from other courts, a new syntax is needed capable of expressing shared and unique attributes and behaviours of instances.

As instances have unique characteristics, depicting these in a conceptual model to communicate their special importance and status becomes more relevant. This is a capability that instance-based modelling is uniquely positioned to perform.

##### 4.2. Case 2: promoting unanticipated uses

As class-based abstractions are created for some purpose(s), these purposes guide the design of a class schema. Class-based conceptual modelling, therefore, limits the flexibility to use data for unanticipated purposes. In contrast, instances in the physical world exist independent of human observers. Therefore, instance representations are less constrained by predefined purposes than are classes. Indeed, the same instance in the real world (e.g., book) can be used for a variety of purposes (e.g., for reading or hold a door open). Therefore, modellers can leverage the innate capabilities of instance-based modelling to promote unanticipated uses of the resulting data.

An example in this case is internet search. Assume an instance represents a unique internet resource (e.g., a URL). To build a general-purpose search engine, it is necessary to be agnostic of the way this unique data is to be used. Consequently, representation of this search engine is merely a set of URLs (instances) having unique attributes. This is the basis for the original implementation of Google's Internet search, which implements a document-based data model (Chang et al., 2008). Google's approach to search contrasts with Yahoo's earlier approach to organising web resources that classified each URL into a predefined set of categories (Spink & Zimmer, 2008).

A conceptual model for this application might represent the top five most popular internet resources as nodes with attributes (see Figure 2). By automatically connecting nodes with the same attributes, business users may visually observe any similarities among most popular resources. Dynamic filters may

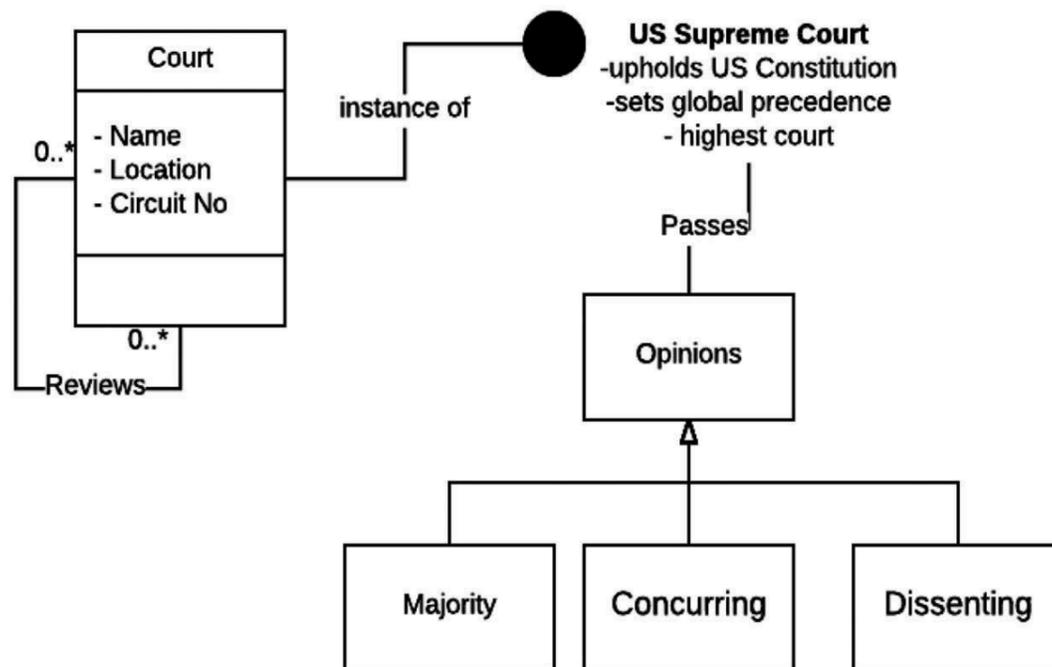


Figure 1. Model of US judicial system using UML-like notation.

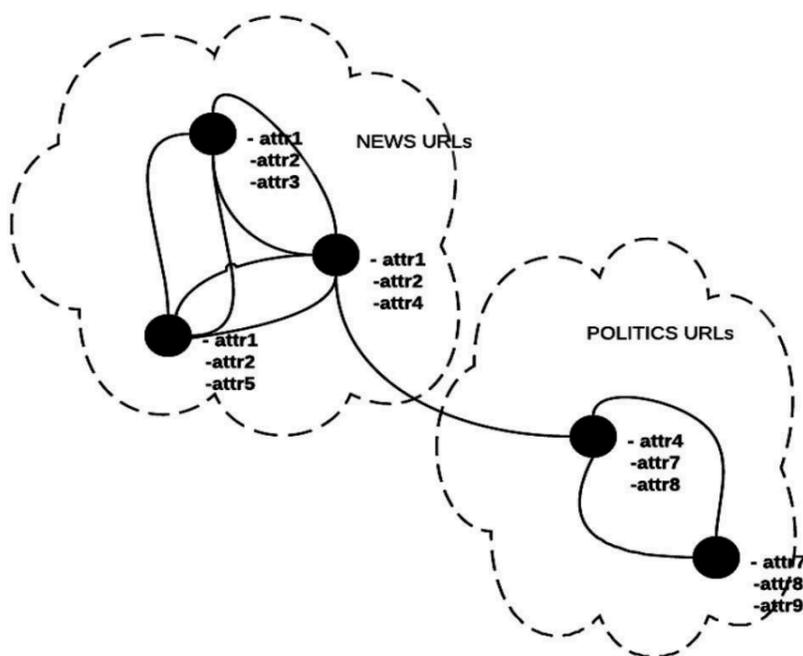


Figure 2. Model of a hypothetical internet search application. (Note: classes are depicted as dashed clouds to indicate their derived nature).

also be used to filter attributes based on situational needs (e.g., URLs with the ability to render on mobile device browsers). Classification structures may also be imposed dynamically reflecting varying needs (see e.g. the dashed lines with labels for the URLs grouped together). In contrast to other types of conceptual models discussed, the primary intent of instance-based models in this case is to support visualisations over existing data. However, as in other cases, a different modelling grammar is needed. In this scenario, it would be important to convey that the classification structures and filters over the data are a layer on top of the instances (Parsons & Wand, 2000). This may require a new way to represent a class construct (by stressing its derived nature, as in Figure 2). Such treatment of classes is meant to stress that any additional abstractions imposed are situational, rather than intrinsically driven by the original domain, and suggests that other ways of partitioning data are possible and, perhaps, should be explored.

Providing an instance-based grammar can also support analysing data to generate actionable insights from modern databases. Modern databases that store and process large volumes of heterogeneous data are increasingly being mined to discover unanticipated patterns (Chang et al., 2008; DeCandia et al., 2007; Hewitt, 2010). To support this, databases and storage technologies (e.g., Hadoop file system, NoSQL) switch from the traditional schema-on-write to the schema-on-read philosophy, in which ad hoc schema is constructed upon data retrieval (Dong & Halevy, 2005; Dong & Srivastava, 2013; Markl, 2014).

One of the goals of the recent wave of data analytics is to look for insights by combining different data sets (e.g., from different IS, from different contexts, and/or obtained from outside the organisation) (Chen et al., 2012; LaValle, Lesser, Shockley, Hopkins, & Kruschwitz, 2014; Minelli, Chambers, & Dhiraj, 2012). In these contexts, it can be dangerous to run automatic routines on data without fully understanding the semantics of the data (Provost & Fawcett, 2013). However, individuals who are analysing the data are not likely to have been involved in the analysis and design of the original IS (Browne & Ramesh, 2002; Davenport & Patil, 2012). Development of models capable of faithfully conveying nuances of data (which, as argued earlier need to be capable of representing individual instances) appears to be instrumental for progress in making this unanticipated use of data beneficial for organisations.

#### 4.3. Case 3: representing and promoting open domain boundaries

As discussed earlier, imposing class-based modelling creates domain boundaries. There could be many practical situations, however, when it is important to be explicit about the open nature of the domain.

As instances stand for individual unique objects, they are incapable of depicting boundaries of the domain. Instance-based conceptual modelling provides a natural capability to represent and convey domain openness.

To illustrate, consider a scenario in which developers wish to create a crowdsourcing IS for disaster response (Goodchild & Glennon, 2010; Majchrzak & More, 2011). This system is intended to be used by search and rescue volunteers—ordinary people who may notice something unusual in their surroundings—leading to a more focused search and rescue effort. As there could be a variety of clues that may lead to a successful search and rescue, fixating volunteers on predefined classes could prove detrimental because each search and rescue operation will be unique. One option is to model the domain of interest for such an application using examples (or instances as exemplars). These communicate typical objects that may be of interest to the emergency response units, but do not restrict volunteers to only those classes of objects and serve as merely illustrations. One can conceive of an instance-based conceptual model containing photographs or stylised representations of typical objects, possibly organised around some higher level categories. This model can be used to train volunteers or can become an input for application development as it signals the development team to keep the user interface and data collection choices as open and flexible as possible.

#### 4.4. Case 4: promoting discoveries

As noted before, class-based conceptual modelling assumes the abstractions needed for a domain are relatively well-defined and stable. This assumption may not be problematic for traditional IS applications within the boundaries of an organisation, where legal requirements, organisational norms, and business processes impose structure and discipline on the requirements. Indeed, the regularity and predictability of well-defined, stable classes is consistent with a view of the organisation in which repeatable activities involving agreed-on classes of entities form the basis of transactions that need to be recorded.

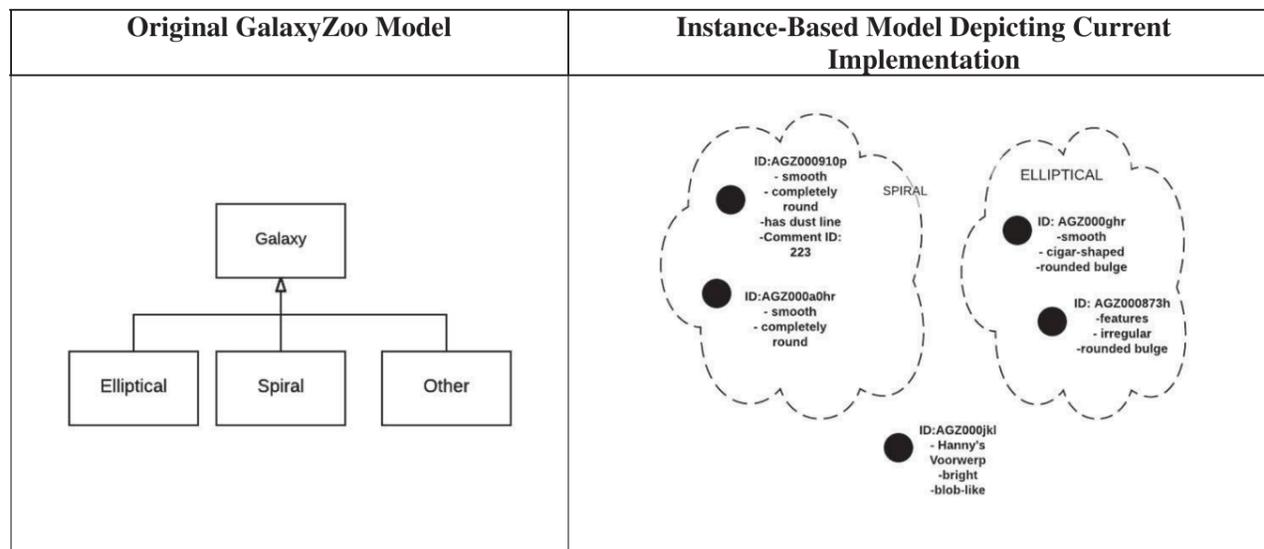
While such applications are critical to organisational success, an emerging class of applications based on connecting the organisation with its external environment is increasingly important, as the desire to monitor and respond to changes in the environment grows. Among these is crowdsourcing, which enables organisations to harness the knowledge and creativity of millions of people across the globe (Brabham, 2013; Doan, Ramakrishnan, & Halevy, 2011; Franklin, Kossmann, Kraska, Ramesh, & Xin, 2011). One use for crowdsourcing is to harness the ingenuity and creativity of ordinary people to

promote discovery in the social or natural environment. In this space, many projects appear to shift from the traditional modelling philosophy and embrace instances (albeit without support from instance-based grammars).

Consider a prominent scientific crowdsourcing project, Zooniverse (zooniverse.org), the world's largest citizen science platform with almost a million users (Simpson, Page, & De Roure, 2014). It originally implemented traditional class-based modelling. Specifically, this meant that when users were shown images of galaxies, there were asked to classify them as one of the predefined categories (which were known and of interest to scientists) (Borne & Team, 2011; Fortson et al., 2011; Simpson et al., 2014; Smith, Lynn, & Lintott, 2013). Very quickly, however, users turned to an online forum to report things that did not fit within the predefined schema, including a new category of galaxies unknown to science (Cardamone et al., 2009) and a novel celestial body coined “voorwerp” (or “object” in Dutch, since it was discovered by a Dutch school teacher) dubbed one of the most exciting recent discoveries in astronomy (Lintott et al., 2009).

The realisation that ordinary people looking at photographs may lead to fundamental scientific breakthrough and even to changes to major classification schemes of astronomy, precipitated a recent change in the design philosophy on Zooniverse. The project adopted an instance-based design philosophy, now asking volunteers to describe instances using visual attributes (although these are still predefined), and adding a mechanism for flagging anything unusual or unexpected via an embedded textbox (Bowyer, Lintott, Hines, Allen, & Paget, 2015). The two models are illustrated in Figure 3. Unlike the original model, the new GalaxyZoo design embraces the possibility of discovering new phenomena and thus capitalises on the potential of crowds better.

Like the new GalaxyZoo, unstructured data collection mechanisms (e.g., loosely structured free-form text, open tags, key-value pairs) are essential in social media and social networking (e.g., Facebook, Twitter, PatientsLikeMe), and are also the primary means of collection and storage in online forums, chats, wikis, and knowledge sharing communities (Bifet & Frank, 2010; Haklay & Weber, 2008; Kallinikos & Tempini, 2014; Russell, 2013; Wattal, Schuff, Mandviwalla, & Williams, 2010). However, no agreed-on approach exists for supporting these developments through conceptual modelling, resulting in a major development vacuum. As one case study reports, while developers were knowledgeable and had expertise in traditional conceptual modelling, they faced considerable uncertainty when they chose to develop a project in a more flexible way, as it was not clear what kind of conceptual modelling techniques and approaches to use (Lukyanenko & Parsons, 2013b).



**Figure 3.** GalaxyZoo’s conceptual models before and after based on Bowyer et al. (2015) and Hopkin (2007).

(Note: Simplified model to show relevant differences; the new model contains realistic data)

#### 4.5. Case 5: instance-based modelling for storage and retrieval

While class-based conceptual modelling is consistent with relational database design, object-oriented programming and form-based user-interfaces, modern technologies in storage, data presentation and data capture appear to be more consistent with the idea of representing instances.

There has been a proliferation of flexible, schema-less, NoSQL databases such as key-value store (DeCandia et al., 2007), column-store (Stonebraker et al., 2005), document-focused (Chang et al., 2008), instance-based (Parsons & Wand, 2000) and graph (Angles & Gutierrez, 2008) data models. A major characteristic of these models is lack of rigid predefined structure (i.e., schema on read), which makes them suitable for storing instances.

Despite developments in practice, the conceptual layer to support flexible database and user interface design is not established. Notably, several attempts to provide such a conceptual layer have been recently made by practitioners (Frisendal, 2016; Hills, 2016). Yet, researchers doubt “the usefulness of data models” for these storage technologies (Kaur & Rani, 2013, p. 3). We argue the opposite: development using previous data models (e.g., relational, network) was supported through the introduction of a conceptual layer in the form of ER or other grammars. Some practitioners argue the need for a conceptual layer is even more pressing in the age of NoSQL databases and Big Data, as it is becoming significantly more challenging to interpret complex and variable interrelationships among data items (Harrison, 2015, p. 197). Instance-based modelling makes it potentially easier to retrieve, visualise, and analyse data stored in modern databases, as we explain below.

While data models support storage and retrieval, conceptual models strive to represent application domains—a need that exists whether or not development relies on flexible storage/presentation.

Although NoSQL databases are typically developed without prior conceptual modelling (Kaur & Rani, 2013), in many cases the aggregates (that is, records stored together on the same physical cluster) can be conceptualised as instances. For example, in Google’s BigTable, aggregates may be individual web pages or web sites, while in Facebook’s Cassandra they are individual user profiles (see Chang et al., 2008; Hewitt, 2010). Relational databases typically store data related to the same individual in different tables; NoSQL databases tend to aggregate data that tends to be retrieved together—this often is data on the same instance. Thus, instance-based models can potentially support data scientists’ understanding of the data available to them and help them manage effective access to it. Because conceptual models can serve as documentation that can be used for future changes to an IS (i.e., maintenance) (Wand & Weber, 2002), instance-based conceptual models can also be utilised as documentation by individuals that might want to analyse the data stored in the NoSQL systems (Rizzi, 2007; Yogev, Roitman, Carmel, & Zwerdling, 2012).

#### 4.6. Case 6: guiding choice of data storage technologies

In addition to providing the conceptual layer for non-relational databases, instance-based conceptual modelling should also guide the selection of the database technology. Traditionally, conceptual models communicate knowledge about the application domain and shape the deep structure of an IS. For example, classes in a conceptual model conventionally are converted to tables in a relational database. These classes also become objects in object-oriented programming; they also inform menu items, sequences of pages and user interface elements. However, conventionally conceptual models did not guide the choice of the

later technologies (e.g., a relational data model and relational DBMS were assumed as choices for the logical layer and storage).

In contrast, with the advent of NoSQL databases that now coexist with relational databases, developers face a choice of storage technology. Since the storage technology is meant to represent the corresponding real-world objects, conceptual modelling, as the activity most closely related to the analysis of the real world, should offer guidance in the selection of storage technology. To illustrate, consider two hypothetical instance-based models shown in Figure 4. Model 1 shows a domain in which instances tend to share attributes and engage in a variety of relationships among each other (e.g., fire ants building a mound, a bird colony, close-knit family). In contrast, Model 2 shows a domain in which instances are quite dissimilar from one another and rarely interact with one another (e.g., distant participants in a supply chain). One can speculate that different storage solutions may be appropriate for the two domains (e.g., a graph database for Model 1, and a document-based database for Model 2). An analysis of the instance-based conceptual models may be used to determine which storage solution is most effective at representing the semantics of a domain. However, more research is needed to determine the process by which such guidance is determined based on the analysis of a conceptual model.

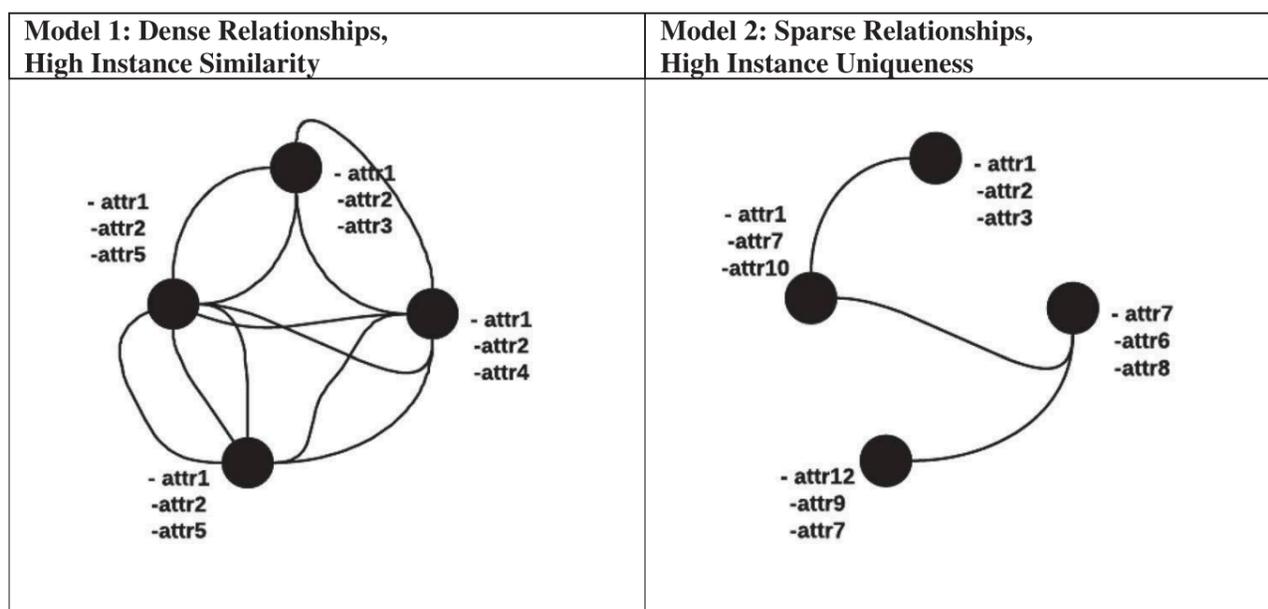
#### 4.7. Case 7: domain understanding from models

Instances are fundamental to the process of establishing mutual understanding between parties involved in IS development. As instances are central to human cognition, instance-based representations can be used to clarify and assimilate abstractions (Anderson et al., 1976; Koedinger, Alibali, & Nathan, 2008), as illustrated in Figure 5.

Representations that emphasise instances in conceptual modelling have shown early promise as being useful to improve domain understanding (Hvalshagen & Samuel, 2014; Samuel, 2012b; Samuel et al., 2018). Similarly, prior research has encouraged the use of instances when creating models for better quality class-based models. Srinivasan and Te'eni (1995) found that individuals who frequently made orderly transitions between classes and instances were able to create a more accurate model than those who did not. Dey, Abowd, and Salber (2001) recommended that individuals could use instances from the real world to analyse the appropriateness of the relationships in class-based models, specifically the number of entity types involved in a relationship in a model. Similarly, Sedrakyian, Snoeck, and Poelmans (2014) built a prototyping tool that helps modellers understand the implications of their modelling decisions on software using instances created by the modeller. Finally, the creation of meaningful entity types should be derived from instances (Parsons, 1996; Parsons & Wand, 1997b). Yet while prior research in IS has, in general, supported the notion of instance-based models, we are not aware of efforts to propose an instance-based *grammar* that would be appropriate for creating these models while facilitating understanding, nor of any sustained effort to demonstrate how such models can resolve the limitations of class-based models.

#### 4.8. Summary of the benefits of instance-based modelling

In this section, we have discussed seven cases we believe instance-based conceptual models are better positioned to handle than class-based models. In Table 2, we provide a potential mapping of which



**Figure 4.** Two hypothetical instance-based conceptual models of different domains. (Note: arcs represent associations).



**Figure 5.** Examples of using instances to support abstraction-based representations and promote shared understanding. (Photos taken by one of the authors on the campus of the University of British Columbia, Canada on July 7 2017).

limitations of class-based models (second column in Table 2) are addressed by the beneficial cases of instance-based models (first column in Table 2). We note that each case addresses one or more limitation and that some of the limitations are directly addressed by a case, whereas others are indirectly addressed by a case. We are encouraged by the prospects instance-based conceptual models offer to help overcome some of the known limitations of class-based models identified earlier in this paper.

## 5. Discussion and implications for research

Notwithstanding the long history of conceptual modelling research, existing conceptual modelling grammars are overwhelmingly based on representation by class-based abstraction. In this paper, we make the case for a more balanced approach that considers instances and instantiation in representations.

To appreciate the potential of instance-based modelling, it is important to understand advantages and limitations of the prevailing class-based conceptual modelling approach that has been practiced for over 40 years. As we discussed and illustrated in this paper, in a class-based modelling scenario it is difficult to achieve stakeholder consensus on how classes should be defined (which attributes are relevant), as different stakeholders might be interested in different attributes of phenomena. This might result in quite complex class definitions, where the final definition does not align with the view of any particular stakeholder. In contrast, a focus on instances recognises that instances may be considered independent of any

classification and can be represented accordingly. This can facilitate alternative classifications by focusing on different subsets of attributes according to a stakeholder's needs.

Because IS development activities, such as database design and interface design, are driven by the conceptual model, in a class-based modelling scenario it is necessary to determine in advance how data are expected to be used and hence identify classes for the model that are congruent with the intended uses. However, instance-based modelling does not rely on a predetermined and fixed classification and development activities are undertaken in the absence of a fixed class-based schema (Lukyanenko et al., 2016). Thus, systems developed this way can be expected to be more easily adapted to evolving uses.

Finally, in class-based conceptual modelling, it is necessary to determine in advance which classes are relevant and select these to be included in the model. This results in a rigid schema that cannot be changed easily if new classes are needed or existing ones are no longer relevant (Parsons & Wand, 2000). Moreover, this schema is reflected in other design objects. However, a modelling approach that incorporates instances allows for classes to be added to or removed from a model as needed. This flexibility arises from the separation of modelling from design activities, in the sense that design objects do not embed the classes of a conceptual model.

Instance-based representation further promises to improve domain understanding based on conceptual models. This kind of grammar also has the potential

**Table 2.** Mapping benefits of instance-based conceptual models to the class-based limitations addressed (Note: Limitation descriptions are only listed where they first appear to save space).

Instance-based conceptual model cases	Class-based limitations addressed
Case 1: Representing Instance Individuality	<ul style="list-style-type: none"> <li>• Limitation 1.1: Class-based models are generally unable to represent uniqueness of instances.</li> <li>• Limitation 1.3: The purpose(s) for the class-based model should be relatively stable over time.</li> <li>• Limitation 2.1: Class-based model may act as anchors, which may bias user's perception of phenomena.</li> <li>• Limitation 4.1: Class-based models may offer distorted impression of instances if inferences drawn are inappropriate for all instances.</li> </ul>
Case 2: Promoting Unanticipated Uses	<ul style="list-style-type: none"> <li>• Limitation 1.2: The purpose(s) for which the class-based model is constructed must be established in advance.</li> <li>• Limitation 1.3</li> </ul>
Case 3: Representing and Promoting Open Domain Boundaries	<ul style="list-style-type: none"> <li>• Limitation 1.1</li> <li>• Limitation 2.1</li> <li>• Limitation 3.1: Class-based models may set domain boundaries that may prevent representation of classes and instances that do not fit within the boundaries.</li> <li>• Limitation 3.2: Class-based models set domain boundaries that may inhibit user engagement.</li> <li>• Limitation 4.1</li> </ul>
Case 4: Promoting Discoveries of Novel Instances and Classes	<ul style="list-style-type: none"> <li>• Limitation 1.1</li> <li>• Limitation 1.2</li> <li>• Limitation 1.3</li> <li>• Limitation 3.2</li> <li>• Limitation 4.1</li> </ul>
Case 5: Modelling for Instance-based Storage and Retrieval	<ul style="list-style-type: none"> <li>• Limitation 1.1</li> <li>• Limitation 1.2</li> <li>• Limitation 1.3</li> </ul>
Case 6: Guiding Choice of Data Storage Technologies	<ul style="list-style-type: none"> <li>• Limitation 1.1</li> <li>• Limitation 1.2</li> <li>• Limitation 1.3</li> </ul>
Case 7: Promoting Understanding of Conceptual Models	<ul style="list-style-type: none"> <li>• Limitation 1.2</li> <li>• Limitation 1.3</li> <li>• Limitation 2.1</li> <li>• Limitation 4.1</li> <li>• Limitation 5.1</li> </ul>

to align better with the rapid growth of flexible database technologies and increased demand to process and analyse massive volumes of heterogeneous data that do not conform to a predetermined schema. Finally, the development of an instance-based grammar may support the inherently heterogeneous and variable user input on social media and avoid the potentially negative consequences of class-based (abstractions) modelling on data quality. Following these arguments, we hope to begin addressing the inherent limitations of class-based modelling through the introduction of formalisms that capture individual instances. *We thus identify developing and evaluating instance-based conceptual modelling grammars*

*as an important future direction for conceptual modelling research.*

However, the shift from class-based representations to those that have greater support for instances is not straightforward. Reflecting some of the limitations of class-based modelling discussed here, practitioners have recently begun to explore instance-based representations (Frisendal, 2016; Hills, 2016). Unfortunately, these efforts are largely occurring in isolation from decades of theoretical and empirical work on conceptual modelling and are currently not supported by academic research. This makes it unclear whether the proposed or adopted and extended grammars are effective at providing the features required by instance-based conceptual modelling. What is currently missing is the academic research component that can develop and evaluate design choices and investigate and support the fledging interest of practitioners in adopting instance-based modelling grammars. We outline key challenges for research that need to be resolved on the path to an effective instance-based conceptual modelling grammar capable of realising the advantages of using instances while addressing the limitations of traditional class-based models.

### 5.1. Challenge 1: how to identify instances and define the instance construct

A fundamental question in how to design an instance-based grammar is having clear and uncontroversial notions of objects, classes, concepts and identity that underlie the instance construct (Eriksson & Agerfalk, 2010; Henderson-Sellers, 2015; Henderson-Sellers, Eriksson, & Ågerfalk, 2015; Parsons & Wand, 2008; Smith, 2004). Despite a long philosophical tradition and strong intuition for the concept of instances, objects, particulars or individuals, we lack a generally agreed on definition of instances suitable for use in conceptual modelling.

For example, should instances be limited to common sense “middle-level” physical objects and/or social entities (Bergholtz & Eriksson, 2015)? Should we only limit instance-based grammars to instances of substance and form, or can we extend the notion of instances to the realm of actions, possibilities and events (Pentland et al., 2017)? As we highlighted in this paper, researchers have sought the answer to this question in underlying theoretical foundations such as psychology, philosophy or linguistics. Yet even adopting a particular ontological position (e.g. the philosophy of Mario Bunge), does not appear to resolve this issue completely, as whether the ontology is only concerned with material or also social objects has been repeatedly debated (Wand & Weber, 2006; Wyssusek, 2006). Bunge's own conception of instances (things) appeared to have evolved over the

years from a more materialistic notion to a broader one, encompassing social objects (Bunge, 1977, 2003).

In addressing this issue, one should also be cognizant of the ease with which this concept can be understood and consistently applied by analysts, designers and users. While theoretical work in this direction is obviously needed, there is also a need for empirical research to ensure that theoretical accounts and arguments translate into empirically validated operationalisations in the context of conceptual modelling.

### 5.2. Challenge 2: how to represent instances

In addition to the search for an effective and unambiguous definition of an instance, a further question is how to represent instances in conceptual models. Traditionally, data representations have primarily relied on language or simple graphics (e.g., diagrams) (Moody, 2009). Yet an important characteristic of instance-based conceptual modelling grammars is that they purport to represent real world objects, which are assumed to be inherently unique and, with the exception of social objects, non-linguistic in nature.

Drawing inspiration from tangible and vivid examples of recyclable items in Figure 5, the question becomes whether the grammar should take advantage of novel forms of rich representation (such as pictures, video and sounds) that can capture the richness of instances. This is consistent with the essence of depicting individual objects (rather than abstractions), and arguments from philosophy on the impossibility of fully representing an instance using language or symbolic graphical forms (Floridi, 2010; Heidegger, 1996; Lukyanenko & Parsons, 2011). Furthermore, with the continued progress in computer-aided software engineering, many conceptual models are no longer drawn in pencil and paper but instead are designed using software. This makes it easier to insert images, and link to textual narratives or pictures and videos. It is conceivable that the search for an effective instance-based representation may foster the rethinking of what a conceptual model should look like.

However desirable the new formats may be, future research should also consider one potential limitation of using instances. This could have an unintended side effect of anchoring (Allen & Parsons, 2010; Tversky & Kahneman, 1974), in which users of models assume that other instances will be very similar to the instances depicted in a model. The broad problem is that existing research offers little guidance in going beyond the traditional text and diagrams. Masri (2009, 2008) demonstrated that use of icons (as opposed to words and geometric shapes) can improve domain understanding. Moody (2009) discussed a variety of approaches to building novel notations, including considerations

of shape, colour and texture. However, the idea of employing rich media in conceptual models has not been considered in great detail by the conceptual modelling community.

### 5.3. Challenge 3: how to use instances to analyse and capture application requirements

While many existing formalisms—such as predicate calculus, conceptual graphs, and RDF—have support for reasoning about and representing individuals, we have little understanding of whether and how incorporating instance-based representations in conceptual modelling grammars can be useful in addressing traditional objectives of conceptual modelling. Unlike traditional class-based grammars that can be used (at least in principle) to capture all relevant facts about the domain, one cannot rely on instances to provide a complete specification of application requirements due to the problem of induction. Given this issue, work is needed to understand what is the most effective use of instances for IS requirements analysis and representation. Potential solutions could involve representing prototypical instances, outliers, or using instances as examples. Another potential possibility is instance sampling, whereby an analyst samples potential instances in a systematic way to minimise bias. These instances are then depicted in a diagram and are meant to represent the domain population. Much of this is an uncharted territory for conceptual modelling research. A promising opportunity in this direction lies in the analysis of emerging approaches for utilising instance-based representations in practice. In adopting such general formalisms as directed graphs, practitioners will undoubtedly face the issue of the impossibility of providing definitive domain boundaries using instances. Researchers should study solutions from practice as this may also provide input into the theoretical and empirical conceptual modelling research.

### 5.4. Challenge 4: how to represent classes in an instance-based grammar

A particularly challenging question deals with the representation of classes and whether a grammar should be solely focused on representing instances or be a hybrid that represents classes as well. To stimulate more research in this area, in this paper we offered a number of advantages of class-based modelling for reasoning and thinking about domains, and for building conceptual models. It is clear that a grammar that represents instances alone would struggle to realise many of the important benefits of class-based models. This is a notable point, as some emerging solutions in practice employ instance-based

representations only (see, Frisendal, 2016). However, this does not preclude the possibility of using pure instance-based approaches for certain situations. Future research should explore the issue of whether and how to mix instances and class-based approaches to accrue maximal benefit from each. Some guidance in addressing this question might come from psychology research on representations that compared the advantages and limitations of using instances in problem solving, thinking and reasoning (Anderson et al., 1976; Koedinger et al., 2008; Koedinger & Nathan, 2004; Markman, 1991; Markman & Ross, 2003; Markovits & Vachon, 1990; Moreno, Ozogul, & Reisslein, 2011).

This challenge also has an important pragmatic dimension. Many real-world domains contain relatively fixed, stable, and agreed-upon classes, attributes, and relationships, making these domains more amenable to class-based modelling. For example, many legal, security and reporting norms follow fixed conventions that need to be captured as universal rules (i.e., abstractions), rather than as a collection of examples (however, there is an important role for individual cases as precedents in law). Similarly, a requirement to exchange data with legacy systems may suggest pre-specifying some structures in advance. At the same time, the challenge in dealing with classes is in their potential to constrain instances (Parsons & Wand, 2000). An open question therefore is what kind of semantics is best captured using class-based versus instance-based representations, considering not only general theoretical issues, but also pragmatic ones?

### 5.5. Challenge 5: how to evaluate and apply instance-based grammars

Once established, an instance-based grammar needs to be rigorously evaluated, which includes assessing its additional utility in IS projects. This work should include developing and evaluating new grammars, as well as evaluating instance-based grammars used in practice. We found cases of grammars used by practitioners in a manner consistent with the ideas expressed in this paper (discussed earlier). To our knowledge, no research has conducted an evaluation of these grammars; therefore, there are exciting opportunities for future studies.

It is notable that much IS development has been conducted using class-based grammars, sometimes relying on multiple models (Dobing & Parsons, 2006; Fettke, 2009; Sabegh, Recker, & Green, 2016), or even eschewing formal modelling entirely (Lukyanenko & Parsons, 2013a; Recker, 2015). Some research has proposed and implemented a “no conceptual modelling” approach, where development skips the modelling stage and relied on a flexible of data model to capture and

store data (Lukyanenko et al., 2017). Similarly, “no modelling” appears to be the default in many NoSQL-type projects (Kaur & Rani, 2013). Another approach focuses on building simple “agile” models, allowing for schema expansion and evolution (Chen, 2006; Liddle & Embley, 2007; Roussopoulos & Karagiannis, 2009). Many popular social media projects continue to implement traditional class-based modelling (Lukyanenko et al., 2014a, 2017; Wiggins et al., 2013). Thus, while we advocate for the construction of an instance-based grammar, it is important to note that alternatives (including the “no grammar” option) should also be considered and the resulting grammar should be gauged against these alternatives as well as traditional class-based modelling.

## 6. Conclusion

Conceptual modelling research and its reference disciplines have developed an extensive body of knowledge on the advantages and shortcomings of design choices in the context of the important activity of conceptual modelling (e.g., Allen & March, 2012; Batra, 2007; Bodart, Patel, Sim, & Weber, 2001; Burton-Jones & Meso, 2008; Jabbari Sabegh, Lukyanenko, Recker, Samuel, & Castellanos, 2017; Moody, 2009; Mylopoulos, 1992; Peckham & Maryanski, 1988; Saghafi & Wand, 2014; Samuel et al., 2018; Shanks, Tansley, & Weber, 2004; Soffer & Hadar, 2007). However, existing conceptual modelling approaches emphasise class-based abstractions while failing to recognise its limitations established from the same reference disciplines used to establish its prominence. Furthermore, with the advent of new database and related information technologies, the efficacy of class-based conceptual modelling has been questioned. In this paper, we suggest instance-based conceptual models be further considered by researchers to counterbalance and alleviate the challenges to class-based models.

In building a new instance-based conceptual modelling grammar, a challenge is synthesising and reconciling prior knowledge to develop effective design solutions. Therefore an opportunity exists to create a grammar that is based on many years of theoretical research. In addition, building a grammar grounded in existing theory-based design principles can be used to evaluate existing theoretical research once the grammar is put to use. The development of an instance-based conceptual modelling grammar will open a new and exciting chapter in conceptual modelling research and practice. But before this story can be written, there are serious challenges to be answered. We thus call upon the community of conceptual modelling researchers to contribute and support this important work.

## Notes

1. In this paper, we use the term “model” to refer to the output of conceptual modelling. Models are also known as diagrams, representations, schemas or scripts (Olivé, 2007; Peckham & Maryanski, 1988; Wand & Weber, 2002).
2. Conceptual modelling research typically uses terms such as (1) classes or entity types, (2) instances, objects, members and entities, and (3) attributes to denote what psychology may refer to as (1) concepts, categories, and classes (2) objects, individuals and (3) features, attributes or properties, respectively. These terms in philosophy are, correspondingly, (1) kinds, universals, sets or classes, (2) things, particulars, objects or individuals, and (3) properties, qualia, or features. For consistency, in this work, we use terms classes and instances when discussing issues related to IS and conceptual modelling, but also use other terms (e.g., objects, individuals) when it is more appropriate to the context or references discussed.
3. See, <http://pubs.opengroup.org/architecture/archimate3-doc/chap09.html>.
4. One workaround to deal with this challenge is to create a class for each new instance, but this would result in the vast increase in complexity of models and proliferation of classes, which would negate the benefits of class-based modelling. Furthermore, this would result in the confusion between what is a class and what is an instance.
5. A set of necessary and sufficient conditions that definitively determined whether a particular *thing* in a domain is a member of the class.

## Disclosure statement

No potential conflict of interest was reported by the authors.

## References

- Abbasi, A., Zhou, Y., Deng, S., & Zhang, P. (2018). Text analytics to support sense-making in social media: A language-action perspective. *MIS Quarterly*, 42(2), 1–38.
- Aggarwal, N., Bhatia, S., & Misra, V. (2016). *Connecting the dots: Explaining relationships between unconnected entities in a knowledge graph*, presented at the International Semantic Web Conference, Springer, pp. 35–39.
- Aguirre-Urreta, M. I., & Marakas, G. M. (2008). Comparing conceptual modeling techniques: A critical review of the EER vs. OO empirical literature. *ACM SIGMIS Database*, 39(2), 9–32.
- Allen, G., & Parsons, J. (2010). Is query reuse potentially harmful? Anchoring and adjustment in adapting existing database queries. *Information Systems Research*, 21(1), 56–77.
- Allen, G. N., & March, S. T. (2012). A research note on representing part-whole relations in conceptual modeling. *MIS Quarterly*, 36(3), 945–964.
- Anderson, R. C., Pichert, J. W., Goetz, E. T., Schallert, D. L., Stevens, K. V., & Trollip, S. R. (1976). Instantiation of general terms. *Journal of Verbal Learning and Verbal Behavior*, 15(6), 667–679.
- Angles, R., & Gutierrez, C. (2008). Survey of graph database models. *ACM Computing Surveys*, 40(1), 1–39.
- Appan, R., & Browne, G. J. (2010). Investigating retrieval-induced forgetting during information requirements determination. *Journal of the Association for Information Systems*, 11(5), 250–275.
- Appan, R., & Browne, G. J. (2012). The impact of analyst-induced misinformation on the requirements elicitation process. *MIS Quarterly*, 36(1), 85–106.
- Atzeni, P., Jensen, C. S., Orsi, G., Ram, S., Tanca, L., & Torlone, R. (2013). The relational model is dead, SQL is dead, and I don't feel so good myself. *ACM SIGMOD Record*, 42(1), 64–68.
- Azevedo, C. L., Iacob, M.-E., Almeida, J. P. A., Van Sinderen, M., Pires, L. F., & Guizzardi, G. (2015). Modeling resources and capabilities in enterprise architecture: A well-founded ontology-based proposal for archimate. *Information Systems*, 54, 235–262.
- Badia, A., & Lemire, D. (2011). A call to arms: revisiting database design. *ACM SIGMOD Record*, 40(3), 61–69.
- Barsalou, L. W. (2014). *Cognitive psychology: An overview for cognitive scientists*. Oxfordshire, UK: Taylor & Francis. Tutorial Essays in Cognitive Science Series <https://books.google.com/books?id=3kbrAgAAQBAJ>.
- Batra, D. (2007). Cognitive Complexity in data modeling: Causes and recommendations. *Requirements Engineering*, 12(4), 231–244.
- Batra, D. (2008). Conceptual data modeling patterns: Representation and validation. *Journal of Database Management*, 16(2), 84–106.
- Bergholtz, M., & Eriksson, O. (2015). Towards a socio-institutional ontology for conceptual modelling of information systems. In M. A. Jeusfeld and K. Karlapalem (eds.), *Advances in conceptual modeling* (pp. 225–235). Berlin: Springer.
- Berlin, B., Breedlove, D. E., Raven, P. H., & Hammel, E. A. (2013). *Principles of tzeltal plant classification: An introduction to the botanical ethnography of a mayan-speaking people of highland, chiapas*, Experimental Virology, Elsevier Science. (<https://books.google.ca/books?id=QHPYBAAAQBAJ>).
- Bifet, A., & Frank, E. (2010). Sentiment knowledge discovery in twitter streaming data discovery science. In B. Pfahringer, G. Holmes, & A. Hoffmann (eds.), *Discovery science* (Vol. 6332, pp. 1–15). Berlin: Springer Berlin/ Heidelberg.
- Bodart, F., Patel, A., Sim, M., & Weber, R. (2001). Should Optional properties be used in conceptual modelling? A theory and three empirical tests. *Information Systems Research*, 12(4), 384–405.
- Borne, K., & Team, Z. (2011). the zooniverse: A framework for knowledge discovery from citizen science data. In *AGU Fall Meeting Abstracts*, 1, 0650.
- Bowker, G. C., & Star, S. L. (2000). *Sorting things out: Classification and its consequences*. Cambridge, MA: MIT Press.
- Bowyer, A., Lintott, C., Hines, G., Allen, C., & Paget, E. (2015). Panoptes, a project building tool for citizen science. In *HCOMP 2015*. San Diego, CA: Panos Ipeirotis and Liz Gerber.
- Brabham, D. C. (2013). *Crowdsourcing*. Cambridge, MA: MIT Press.
- Browne, G. J., & Parsons, J. (2012). More enduring questions in cognitive is research. *Journal of the Association for Information Systems*, 13(12), 1000–1011.
- Browne, G. J., & Ramesh, V. (2002). Improving information requirements determination: A cognitive perspective. *Information & Management*, 39(8), 625–645.

- Brynjolfsson, E., Geva, T., & Reichman, S. (2016). Crowd-squared: Amplifying the predictive power of search trend data. *MIS Quarterly*, 40(4), 941–961.
- Bugiotti, F., Cabibbo, L., Atzeni, P., & Torlone, R. (2014). *Database design for NoSQL systems*, presented at the International Conference on Conceptual Modeling, Springer, pp. 223–231.
- Bunge, M. (1977). *Treatise on basic philosophy: Ontology I: The furniture of the world*. Boston, MA: Reidel.
- Bunge, M. (2003). *Emergence and convergence: Qualitative novelty and the unity of knowledge*. Toronto, ON: University of Toronto Press.
- Bunge, M. A. (1979). *Treatise on basic philosophy: Ontology II: A world of systems*. Boston, MA: Reidel Publishing Company.
- Burton-Jones, A., & Meso, P. N. (2008). The effects of decomposition quality and multiple forms of information on Novices' understanding of a domain from a conceptual model. *Journal of the Association for Information Systems*, 9(12), 748–802.
- Burton-Jones, A., Recker, J., Indulska, M., Green, P., & Weber, R. (2017). Assessing representation theory with a framework for pursuing success and failure. *MIS Quarterly*, 41(4), 1307–1333.
- Burton-Jones, A., & Volkoff, O. (2017). How can we develop contextualized theories of effective use? A demonstration in the context of community-care electronic health records. *Information Systems Research*, 28(3), 468–489.
- Burton-Jones, A., & Weber, R. (2014). Building conceptual modeling on the foundation of ontology. In Heikki Topi and Allen Tucker (eds.), *Computing handbook: Information systems and information technology* (pp. 15.1–15.24). Boca Raton, FL, USA: CRC Press.
- Cardamone, C., Schawinski, K., Sarzi, M., Bamford, S. P., Bennert, N., Urry, C., ... Nichol, R. C. (2009). Galaxy zoo green peas: Discovery of a class of compact extremely star-forming galaxies. *Monthly Notices of the Royal Astronomical Society*, 399(3), 1191–1205.
- Carey, S. (2009). *The origin of concepts*. New York, USA: Oxford University Press. [http://books.google.ca/books?id=FBfGNyIk\\_GUC](http://books.google.ca/books?id=FBfGNyIk_GUC)
- Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., ... Gruber, R. E. (2008). Bigtable: A distributed storage system for structured data. *ACM Transactions on Computer Systems*, 26(2), 4–23.
- Checkland, P., & Holwell, S. (1998). *Information, systems, and information systems: Making sense of the field*. Hoboken, NJ: John Wiley & Sons, Inc.
- Chen, H., Chiang, R. H., & Storey, V. C. (2012). Business intelligence and analytics: From big data to big impact. *MIS Quarterly*, 36(4), 1165–1188.
- Chen, P. (1976). The entity-relationship model - toward a unified view of data. *ACM Transactions on Database Systems*, 1(1), 9–36.
- Chen, P. (2006). Suggested research directions for a new frontier – active conceptual modeling. In D. W. Embley, A. Olivé, and S. Ram (Eds.), *ER 2006* (pp. 1–4). Lecture Notes in Computer Science.
- Clarke, R., Burton-Jones, A., & Weber, R. (2016). On the ontological quality and logical quality of conceptual-modeling grammars: The need for a dual perspective. *Information Systems Research*, 27(2), 365–382.
- Colner, R., & Rehder, B. (2009). A new theory of classification and feature inference learning: An exemplar fragment model. In Niels Taatgen and Hedderik van Rijn (Eds.), *Proceedings of the Cognitive Science Society* (Vol. 31, pp. 1–14).
- Davenport, T. H., & Patil, D. (2012). Data scientist. *Harvard Business Review*, 90, 70–76.
- Davies, I., Green, P., Rosemann, M., Indulska, M., & Gallo, S. (2006). How do practitioners use conceptual modeling in practice? *Data & Knowledge Engineering*, 58(3), 358–380.
- DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., ... Vogels, W. (2007). Dynamo: Amazon's highly available key-value store. In *ACM SIGOPS operating systems review* (Vol. 41, pp. 205–220). ACM.
- Dey, A. K., Abowd, G. D., & Salber, D. (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction*, 16(2), 97–166.
- Doan, A., Ramakrishnan, R., & Halevy, A. Y. (2011). Crowdsourcing systems on the world-wide web. *Communications of the ACM*, 54(4), 86–96.
- Dobing, B., & Parsons, J. (2006). How UML is used. *Communications of the ACM*, 49(5), 109–113.
- Dong, X., & Halevy, A. (2005). Malleable schemas: A preliminary report. In *Proc. of WebDB* (pp. 139–144).
- Dong, X. L., & Srivastava, D. (2013). Big data integration. In A. Doan, F. Neven, R. Mccann, & J. Bex (Eds.), *Data engineering (ICDE), 2013 IEEE 29th international conference on* (pp. 1245–1248). IEEE.
- Duncan, J. (1984). Selective attention and the organization of visual information. *Journal of Experimental Psychology: General*, 113(4), 501.
- Elmasri, R., & Navathe, S. (2009). *Fundamentals of database systems*. Boston, MA: AddisonWesley.
- Eriksson, O., & Agerfalk, P. J. (2010). Rethinking the meaning of identifiers in information infrastructures. *Journal of the Association for Information Systems*, 11(8), 433–454.
- Eriksson, O., Johannesson, P., & Bergholtz, M. (2018). Institutional ontology for conceptual modeling. *Journal of Information Technology*, 33(2), 1–19.
- Falkowski, A., & Feret, B. (1990). Prototype and exemplar models in categorization: A simulatory comparative analysis. *Polish Psychological Bulletin*, 21(3), 199–211.
- Feigl, H., Scriven, M., & Maxwell, G. (1958). *Concepts, theories, and the mind-body problem*. Minneapolis, Minnesota: Minnesota Studies in the Philosophy of Science, University of Minnesota Press. <https://books.google.ca/books?id=99WNU7gabN4C>
- Fettke, P. (2009). How conceptual modeling is used. *Communications of the Association for Information Systems*, 25(1), 43.
- Floridi, L. (2010). *Information: A very short introduction*. Oxford, UK: Oxford University Press.
- Fortson, L., Masters, K., Nichol, R., Borne, K., Edmondson, E., Lintott, C., ... Wallin, J. (2011). Galaxy Zoo: Morphological classification and citizen science. In M. J. Way, J. D. Scargle, K. M. Ali, A. N. Srivastava (Eds.), *Advances in Machine Learning and Data Mining for Astronomy* (pp. 1–11). Boca Raton, FL: Chapman & Hall/CRC.
- Foster, J. (2011). Ontologies without metaphysics: Latour, harman and the philosophy of things. In *Analecta hermeneutica* (Vol. 3, pp. 3).
- Franklin, M. J., Kossmann, D., Kraska, T., Ramesh, S., & Xin, R. (2011). "CrowdDB: Answering queries with crowdsourcing," in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of*

- Data*, SIGMOD '11, Athens, Greece: ACM, pp. 61–72. doi:10.1145/1989323.1989331.
- Frisendal, T. (2016). *Graph data modeling for NoSQL and SQL: Visualize structure and meaning*. Basking Ridge, NJ: Technics Publications.
- Germonprez, M., Hovorka, D., & Collopy, F. (2007). A theory of tailorable technology design. *Journal of the Association for Information Systems*, 8(6), 351–367.
- Gilovich, T., Griffin, D., & Kahneman, D. (2002). *Heuristics and Biases: The psychology of intuitive judgment*. Cambridge, UK: Cambridge University Press.
- Gomez-Perez, J. M., Pan, J. Z., Vetere, G., & Wu, H. (2017). Enterprise knowledge graph: An introduction. In *Exploiting linked data and knowledge graphs in large organisations* (pp. 1–14). Springer.
- Goodchild, M. F., & Glennon, J. A. (2010). Crowdsourcing geographic information for disaster response: A research frontier. *International Journal of Digital Earth*, 3(3), 231–241.
- Grolinger, K., Higashino, W. A., Tiwari, A., & Capretz, M. A. (2013). Data management in cloud environments: NoSQL and newSQL data stores. *Journal of Cloud Computing: Advances, Systems and Applications*, 2(1), 22.
- Hahn, U. (2014). Similarity. In *Wiley Interdisciplinary Reviews: Cognitive Science*, 5, 271–280.
- Haklay, M., & Weber, P. (2008). OpenStreetMap: User-generated street maps. *IEEE Pervasive Computing*, 7(4), 12–18.
- Halpin, T. (2007). Fact-oriented modeling: Past, present and future. In J. Krogstie, A. L. Opdahl, & S. Brinkkemper (Eds.), *Conceptual modelling in information systems engineering* (pp. 19–38 342). Berlin: Springer.
- Harnad, S. R. (1990). *Categorical perception: The groundwork of cognition*. Cambridge, MA: Cambridge University Press. <http://books.google.ca/books?id=rG1BnQeT9L4C>
- Harrison, G. (2015). *Next generation databases: NoSQL, NewSQL, and big data, new york*. NY, USA: Apress.
- Heath, T., & Bizer, C. (2011). *Linked data: Evolving the web into a global data space* (Vol. 1)). San Rafael, CA: Morgan & Claypool Publishers.
- Heidegger, M. (1996). *Being and time: A translation of Sein Und Zeit*. New York, NY: State University of New York Press. <http://books.google.ca/books?id=9oc2BnZMCZgC>
- Henderson-Sellers, B. (2015). Why philosophize; why not just model?. In *Conceptual modeling* (pp. 3–17). Springer.
- Henderson-Sellers, B., Eriksson, O., & Ågerfalk, P. J. *On the need for identity in ontology-based conceptual modelling*, In Presented at the proceedings of the 11th Asia-Pacific Conference on Conceptual Modelling (APCCM) (Vol. 27), 2015), p. 30.
- Hewitt, E. (2010). *Cassandra: The definitive guide*. Sebastopol, CA: O'Reilly Media.
- Hills, T. (2016). *NoSQL and SQL data modeling: bringing together data, semantics, and software*. Basking Ridge, NJ: Technics Publications.
- Hirschheim, R., Klein, H. K., & Lyytinen, K. (1995). *Information systems development and data modeling: Conceptual and philosophical foundations*. Cambridge, UK: Cambridge University Press. <http://books.google.ca/books?id=Z4s2IwuyZNUC>
- Hopkin, M. (2007). See new galaxies—without leaving your chair. In *Nature news online* (pp. 11). <https://www.nature.com/news/2007/070709/full/news070709-7.html>
- Hvalshagen, M., & Samuel, B. M. 2014. “Aiding the comprehension of cardinality constraints with narratives,” In *13th Symposium on Research in Systems Analysis and Design* St. Louis, Missouri, USA, pp. 5–14.
- Hvalshagen, M., Samuel, B. M., & Lukyanenko, R. 2017. “Conceptual data models and narratives: A tool to help the tool,” in *SIGSAND Symposium*, Cincinnati, Ohio, pp. 1–10.
- Jabbari Sabegh, M. A., Lukyanenko, R., Recker, J. C., Samuel, B. M., & Castellanos, A. 2017. “Conceptual modeling research in information systems: What we now know and what we still do not know. In *AIS SIGSAND*, Cincinnati, Ohio, pp. 1–12.
- Jacobson, I., Booch, G., & Rumbaugh, J. (1999). *The unified software development process* (Vol. 1). Reading MA: Addison-Wesley.
- Jagadish, H., Gehrke, J., Labrinidis, A., Papakonstantinou, Y., Patel, J. M., Ramakrishnan, R., & Shahabi, C. (2014). Big data and its technical challenges. *Communications of the ACM*, 57(7), 86–94.
- Kahn, B. E., & Sarin, R. K. (1988). Modelling ambiguity in decision under uncertainty. *Journal of Consumer Research*, 15(2), 255–262.
- Kahneman, D. D. (1992). The reviewing of object files: Object-specific integration of information. *Cognitive Psychology*, 24(2), 175–219.
- Kallinikos, J., & Tempini, N. (2014). Patient data as medical facts: Social media practices as a foundation for medical knowledge creation. *Information Systems Research*, 25(4), 817–833.
- Kaur, K., & Rani, R. 2013. “Modeling and querying data in NoSQL databases,” in *Big data, 2013 IEEE international conference on*, IEEE, pp. 1–7.
- Khatri, V., Vessey, I., Ramesh, V., Clay, P., & Park, S.-J. (2006). Understanding conceptual schemas: Exploring the role of application and IS domain knowledge. *Info. Sys. Research*, 17(1), 81–99.
- Kleek, M. G. V., Styke, W., Schraefel, M., & Karger, D. 2011. “Finders/keepers: A Longitudinal study of people managing information scraps in a micro-note tool,” in *Proceedings of the SIGCHI conference on human factors in computing systems*, Vancouver, BC, Canada: ACM, pp. 2907–2916.
- Koedinger, K. R., Alibali, M. W., & Nathan, M. J. (2008). Trade-offs between grounded and abstract representations: Evidence from algebra problem solving. *Cognitive Science*, 32(2), 366–397.
- Koedinger, K. R., & Nathan, M. J. (2004). the real story behind story problems: Effects of representations on quantitative reasoning. *The Journal of the Learning Sciences*, 13(2), 129–164.
- Kroupa, P., Pawlowski, M., & Milgrom, M. (2012). The failures of the standard model of cosmology require a new paradigm. *International Journal of Modern Physics D*, 21(14), 1230003.
- Kung, C. H., & Solvberg, A. 1986. “Activity modeling and behavior modeling,” in *Proc. of the IFIP WG 8.1 Working Conference on Information Systems Design Methodologies: Improving the Practice*, North-Holland Publishing Co., pp. 145–171.
- Lakoff, G. (1987). *Women, fire, and dangerous things : what categories reveal about the mind*. Chicago: University of Chicago Press.
- Lankhorst, M. M., Proper, H. A., & Jonkers, H. (2010). The anatomy of the archimate language. *International Journal of Information System Modeling and Design (IJISMD)*, 1(1), 1–32.

- Larkey, L. B., & Markman, A. B. (2005). Processes of similarity judgment. *Cognitive Science*, 29(6), 1061–1076.
- Larsen, K. R., & Becker, D. S. (Forthcoming). *Automated machine learning for business: An Introduction to accurate, easy, and fast analytics*. Oxford, UK: Oxford University Press.
- LaValle, S., Lesser, E., Shockley, R., Hopkins, M. S., & Kruschwitz, N. (2014). Big data, analytics and the path from insights to value. *MIT Sloan Management Review*, 52(2), 21–31.
- Lavie, N., Hirst, A., De Fockert, J. W., & Viding, E. (2004). Load theory of selective attention and cognitive control. *Journal of Experimental Psychology: General*, 133(3), 339.
- Lee, Y. W. (2003). Crafting Rules: context-reflective data quality problem solving. *Journal of Management Information Systems*, 20(3), 93–119.
- Lee, Y. W., Pipino, L. L., Funk, J. D., & Wang, R. Y. (2006). *Journey to data quality*. Cambridge, MA: MIT Press.
- Levina, N., & Arriaga, M. (2014). Distinction and status production on user-generated content platforms: Using Bourdieu's theory of cultural production to understand social dynamics in online fields. *Information Systems Research*, 25(3), 468–488.
- Liddle, S. W., & Embley, D. W. (2007). A common core for active conceptual modeling for learning from surprises. In P. C. Peter & Y. W. Leah (eds.), *Active conceptual modeling of learning, active conceptual modeling of learning* (pp. 47–56). Berlin: Springer-Verlag.
- Lintott, C. J., Schawinski, K., Keel, W., Arkel, H. V., Bennert, N., Edmondson, E., ... Vandenberg, J. (2009). Galaxy zoo: Hanny's voorwerp, a quasar light echo? *Monthly Notices of the Royal Astronomical Society*, 399(1), 129–140.
- Lukyanenko, R., & Parsons, J. (2011). Unintended consequences of class-based ontological commitment. In O. De Troyer, C. Bauzer Medeiros, R. Billen, P. Hallot, A. Simitsis, & H. Van Mingroot (eds.), *Advances in conceptual modeling. recent developments and new directions*, (Vol. 6999) (pp. 220–229). Berlin/Heidelberg: Springer.
- Lukyanenko, R., & Parsons, J. (2013a). Is traditional conceptual modeling becoming obsolete? In W. Ng, V. C. Storey, and J. Trujillo (Eds.), *Conceptual Modeling*, 1–14.
- Lukyanenko, R., & Parsons, J. (2013b). Reconciling theories with design choices in design science research. In J. vom Brocke et al. (Eds.), *DESRIST 2013, LNCS 7939, lecture notes in computer science* (pp. 165–180). Berlin/Heidelberg: Springer.
- Lukyanenko, R., Parsons, J., & Wiersma, Y. (2014a). The IQ of the crowd: Understanding and improving information quality in structured user-generated content. *Information Systems Research*, 25(4), 669–689.
- Lukyanenko, R., Parsons, J., & Wiersma, Y. (2014b). The impact of conceptual modeling on dataset completeness: A field experiment. In *Proceedings of the International Conference on Information Systems (ICIS)*, pp. 1–18.
- Lukyanenko, R., Parsons, J., Wiersma, Y., Sieber, R., & Maddah, M. (2016). Participatory design for user-generated content: understanding the challenges and moving forward. *Scandinavian Journal of Information Systems*, 28(1), 37–70.
- Lukyanenko, R., Parsons, J., Wiersma, Y. F., Wachinger, G., Huber, B., & Meldt, R. (2017). Representing crowd knowledge: guidelines for conceptual modeling of user-generated content. *Journal of the Association for Information Systems*, 18(4), 297–339.
- Lukyanenko, R., & Samuel, B. M. (2017). Are all classes created equal? increasing precision of conceptual modeling grammars. *ACM Transactions on Management Information Systems (TMIS)*, 40(2), 1–25.
- Majchrzak, A. N. N., & More, P. H. B. (2011). Emergency! Web 2.0 to the Rescue! *Communications of the ACM*, 54(4), 125–132.
- Mannheim, P. D. (2006). Alternatives to dark matter and dark energy. *Progress in Particle and Nuclear Physics*, 56(2), 340–445.
- March, S., & Allen, G. (2012). Toward a social ontology for conceptual modeling. In *11th Symposium on research in systems analysis and design*, Vancouver, Canada, pp. 57–62.
- March, S., & Allen, G. (2015). Classification with a purpose. In *Symposium on research in systems analysis and design*, Richmond, VA, pp. 1–10.
- Markl, V. (2014). Breaking the chains: On declarative data analysis and data independence in the big data era. *Proceedings of the VLDB Endowment*, 7(13), 1730–1733.
- Markman, A. B., & Ross, B. H. (2003). Category use and category learning. *Psychological Bulletin*, 129(4), 592.
- Markman, E. M. (1991). *Categorization and naming in children: Problems of induction*. Cambridge, Mass: MIT Press.
- Markovits, H., & Vachon, R. (1990). Conditional reasoning, representation, and level of abstraction. *Developmental Psychology*, 26(6), 942.
- Masri, K. (2009). "Conceptual Model Design for Better Understanding," PhD thesis, PhD thesis, Vancouver, BC, Canada: Simon Fraser University.
- Masri, K., Parker, D., & Gemino, A. (2008). Using iconic graphics in entity-relationship diagrams: The impact on understanding. *Journal of Database Management (JDM)*, 19(3), 22–41.
- Mattessich, R. (2013). *Reality and accounting: ontological explorations in the economic and social sciences*. Abingdon-on-Thames, UK: Routledge.
- Meertens, L. O., Iacob, M.-E., Nieuwenhuis, L. J., Van Sinderen, M., Jonkers, H., & Quartel, D. (2012). *Mapping the business model canvas to archimate*, Presented at the Proceedings of the 27th annual ACM symposium on applied computing, ACM, pp. 1694–1701.
- Minelli, M., Chambers, M., & Dhiraj, A. (2012). *Big data, big analytics: Emerging business intelligence and analytic trends for today's businesses*. Hoboken, NJ: John Wiley & Sons.
- Moody, D. L. (2009). The 'physics' of notations: Toward a scientific basis for constructing visual notations in software engineering. *IEEE Transactions on Software Engineering*, 35(6), 756–779.
- Moreno, R., Ozogul, G., & Reisslein, M. (2011). Teaching with concrete and abstract visual representations: Effects on students' problem solving, problem representations, and learning perceptions. *Journal of Educational Psychology*, 103(1), 32.
- Mumford, E., & Henshall, D. (1979). *Designing participatively: A participative approach to computer systems design: A case study of the introduction of a new computer system*. Manchester, UK: Manchester Business School.
- Murphy, G. (2004). *The big book of concepts*. Cambridge, MA: MIT Press.
- Mylopoulos, J. (1992). Conceptual Modeling and Telos. In P. Loucopoulos & R. Zicari (eds.), *Conceptual modeling, databases, and CASE: An integrated view of information systems development* (pp. 49–68). New York, NY: John Wiley & Sons, Inc.
- Mylopoulos, J. (1998). Information modeling in the time of the revolution. *Information Systems*, 23(3–4), 127–155.

- Mylopoulos, J., Chung, L., & Yu, E. (1999). From object-oriented to goal-oriented requirements analysis. *Communications of the ACM*, 42(1), 31–37.
- Norman, D. A. (2002). *The design of everyday things*. New York, NY: Basic Books.
- Ogunseye, S., Parsons, J., & Lukyanenko, R. (2017). Do crowds go stale? Exploring the effects of crowd reuse on data diversity. In *WITS 2017*, Seoul, South Korea.
- Olivé, A. (2007). *Conceptual modeling of information systems*. Berlin, Germany: Springer Science & Business Media.
- The Open Group. (2017). *ArchiMate® 3.0.1 Specification*, Van Haren Publishing. (<https://books.google.ca/books?id=Jmo3DwAAQBAJ>).
- Pap, A. (1978). Disposition concepts and extensional logic. In Raimo Tuomela (Ed.), *Dispositions* (pp. 27–54). Springer.
- Parnas, D. L. (1972). A technique for software module specification with examples. *Communications of the ACM*, 15(5), 330–336.
- Parsons, J. (1996). An information model based on classification theory. *Management Science*, 42(10), 1437–1453.
- Parsons, J., & Wand, Y. (1997a). Choosing classes in conceptual modeling. *Communications of the ACM*, 40(6), 63–69.
- Parsons, J., & Wand, Y. (1997b). Using objects for systems analysis. *Communications of the ACM*, 40(12), 104–110.
- Parsons, J., & Wand, Y. (2000). Emancipating instances from the tyranny of classes in information modeling. *ACM Transactions on Database Systems*, 25(2), 228–268.
- Parsons, J., & Wand, Y. (2008). Using cognitive principles to guide classification in information systems modeling. *MIS Quarterly*, 32(4), 839–868.
- Patel-Schneider, P. F., & Horrocks, I. (2007). A comparison of two modelling paradigms in the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(4), 240–250.
- Peckham, J., & Maryanski, F. (1988). Semantic data models. *ACM Computing Surveys*, 20(3), 153–189.
- Pentland, B., Recker, J., & Kim, I. (2017). Capturing reality in flight? Empirical tools for strong process theory. In *ICIS 2017*, Seoul, South Korea, pp. 1–12.
- Pilone, D., & Pitman, N. (2005). *UML 2.0 in a nutshell*. Sebastopol, CA: O'Reilly Media, Inc.
- Provost, F., & Fawcett, T. (2013). *Data science for business: What you need to know about data mining and data-analytic thinking*. Sebastopol, CA: O'Reilly Media, Inc.
- Quine, W. V. (1977). Intensions Revisited. *Midwest Studies in Philosophy*, 2(1), 5–11.
- Rai, A. (2016). Editor's comments: Synergies between big data and theory. *MIS Quarterly*, 40(1), Iii–Ix.
- Ramesh, V., Parsons, J., & Browne, G. (1999). What is the role of cognition in conceptual modeling? A report on the first workshop on cognition and conceptual modeling. *Conceptual Modeling*, pp. 272–280.
- Recker, J. (2015). Research on conceptual modelling: Less known knowns and more unknown unknowns, please. In *Asia-Pacific conference on conceptual modelling* (Vol. 165), pp. 3–8.
- Recker, J., Rosemann, M., Green, P., & Indulska, M. (2011). Do ontological deficiencies in modeling grammars matter? *MIS Quarterly*, 35(1), 57–79.
- Ritvo, H. (1997). *The platypus and the mermaid, and other figments of the classifying imagination*. Cambridge, Mass: Harvard University Press.
- Rizzi, S. (2007). Conceptual modeling solutions for the data warehouse. *Data Warehouses and OLAP: Concepts, Architectures and Solutions*, 1–26.
- Rossi, M., & Siau, K. (2000). *Information modeling in the new millennium*. Hershey, PA: IGI Global.
- Roussopoulos, N., & Karagiannis, D. (2009). Conceptual modeling: Past, present and the continuum of the future. In A. Borgida, V. Chaudhri, P. Giorgini, & E. Yu (eds.), *Conceptual modeling: Foundations and applications* (Vol. 5600, pp. 139–152). Berlin/Heidelberg: Springer.
- Russell, M. A. (2013). *Mining the social web: Data mining Facebook, Twitter, LinkedIn, Google+, GitHub, and more*. Sebastopol, CA: O'Reilly Media, Inc.
- Sabegh, M. A. J., Recker, J., & Green, P. (2016). Designing experiments to test the theory of combined ontological coverage. In *International Conference on Information Systems*, Dublin, pp. 1–12.
- Saghafi, A., & Wand, Y. (2014). Conceptual models? A meta-analysis of empirical work. In *Hawaii International Conference on System Sciences*, Big Island, Hawaii, pp. 1–15.
- Samuel, B. M. (2012a). Conceptual schema cardinality understanding: The effects of different types of information encoding. PhD thesis, PhD thesis, Bloomington, Indiana USA: Indiana University.
- Samuel, B. M. (2012b). Reconceptualizing conceptual schema comprehension: Understanding the role of instantiation and abstraction. *10th Symposium on Research in Systems Analysis and Design*, pp. 21–27.
- Samuel, B. M., Khatri, V., & Ramesh, V. (2018). Exploring the effects of extensional versus intentional representations on domain understanding. *MIS Quarterly*, 1–40. Forthcoming.
- Samuel, B. M., Watkins, L., Ehle, A., & Khatri, V. (2015). Customizing the representation capabilities of process models: Understanding the effects of perceived modeling impediments. *IEEE Transactions on Software Engineering*, 41(1), 19–39.
- Santos, P. S., Jr, Almeida, J. P. A., & Guizzardi, G. (2013). An ontology-based analysis and semantics for organizational structure modeling in the. *Information Systems A. R. I. S. Method*, 38(5), 690–708.
- Schram, A., & Anderson, K. M. (2012). MySQL to NoSQL: Data modeling challenges in supporting scalability. In *Proceedings of the 3rd Annual Conference on Systems, Programming, and Applications: Software for Humanity*, ACM, pp. 191–202.
- Schwering, A. (2005). Hybrid model for semantic similarity measurement. In R. Meersman & Z. Tari (eds.), *On the move to meaningful internet systems 2005: CoopIS, DOA, and ODBASE* (Vol. 3761, pp. 1449–1465). Berlin/Heidelberg: Springer.
- Searle, J. R. (1983). *Intentionality: An essay in the philosophy of mind*. Cambridge: Cambridge University Press.
- Searle, J. R. (1995). *The construction of social reality*. New York City, NY: Simon and Schuster.
- Sedrakyan, G., Snoeck, M., & Poelmans, S. (2014). Assessing the effectiveness of feedback enabled simulation in teaching conceptual modeling. *Computers & Education*, 78, 367–382.
- Shanks, G., Tansley, E., & Weber, R. (2003). Using ontology to validate conceptual models. *Communications of the ACM*, 46(10), 85–89.
- Shanks, G., Tansley, E., & Weber, R. (2004). Representing composites in conceptual modeling. *Communications of the ACM*, 47(7), 77–80.

- Siau, K. (2004). Informational and computational equivalence in comparing information modeling methods. *Journal of Database Management (JDM)*, 15(1), 73–86.
- Simpson, R., Page, K. R., & De Roure, D. (2014). “Zooniverse: Observing the world’s largest citizen science platform,” In *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion*, International World Wide Web Conferences Steering Committee, pp. 1049–1054.
- Smith, A. M., Lynn, S., & Lintott, C. J. (2013). “An Introduction to the Zooniverse,” In *First AAAI Conference on Human Computation and Crowdsourcing*, Palm Springs, California, pp. 1–10.
- Smith, B. (2004). *Beyond concepts: Ontology as reality representation*, Presented at the Proceedings of the third international conference on formal ontology in information systems (FOIS 2004), pp. 73–84.
- Smith, J. M., & Smith, D. C. P. (1977). Database abstractions: Aggregation and generalization. *ACM Transactions on Database Systems*, 2(2), 105–133.
- Soffer, P., & Hadar, I. (2007). Applying ontology-based rules to conceptual modeling: A reflection on modeling decision making. *European Journal of Information Systems*, 16(5), 599–611.
- Soffer, P., Kaner, M., & Wand, Y. (2008). Assigning ontology-based semantics to process models: The case of petri nets. In Z. Bellahsene and M. Léonard (Eds.), *Advanced Information Systems Engineering* (pp. 16–31). Berlin: Springer.
- Soffer, P., Wand, Y., & Kaner, M. (2015). Conceptualizing routing decisions in business processes: Theoretical analysis and empirical testing. *Journal of the Association for Information Systems*, 16(5), 345.
- Sowa, J. F. (1979). Semantics of conceptual graphs. In *Proceedings of the 17th Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, pp. 39–44.
- Sowa, J. F. (1980). A conceptual schema for knowledge-based systems. *ACM SIGMOD Record*, 11(1), 193–195.
- Sowa, J. F. (2000). *Knowledge representation: Logical, philosophical, and computational foundations*. New York NY: Brooks/Cole.
- Sowa, J. F., & Zachman, J. A. (1992). Extending and formalizing the framework for information systems architecture. *IBM Systems Journal*, 31(3), 590–616.
- Spink, A., & Zimmer, M. (2008). *Web search: Multidisciplinary perspectives, information science and knowledge management*. Berlin, Heidelberg: Springer. <https://books.google.ca/books?id=nopJzzYeNREC>
- Srinivasan, A., & Te’eni, D. (1995). Modeling as constrained problem solving: An empirical study of the data modeling process. *Management Science*, 41(3), 419–434.
- Steiner, T., Verborgh, R., Troncy, R., Gabarro, J., & Van de Walle, R. (2012). *Adding realtime coverage to the google knowledge graph*, Presented at the Proceedings of the 2012th International Conference on Posters & Demonstrations Track-Volume 914, CEUR-WS. org, pp. 65–68.
- Stonebraker, M. (2010). SQL databases v. NoSQL databases. *Communications of the ACM*, 53(4), 10–11.
- Stonebraker, M., Abadi, D. J., Batkin, A., Chen, X., Cherniack, M., Ferreira, M., ... Zdonik, S. (2005). “C-Store: A Column-Oriented DBMS,” in *31st International Conference on Very Large Data Bases*, Trondheim, Norway, pp. 553–564.
- Strong, D. M., & Volkoff, O. (2010). Understanding organization-enterprise system fit: A path to theorizing the information technology artifact. *MIS Quarterly*, 34(4), 731–756.
- Taghavi, A., & Woo, C. (2017). The role clarity framework to improve requirements gathering. *ACM Transactions on Management Information Systems (TMIS)*, 8(2–3), 9.
- Topi, H., & Ramesh, V. (2002). *Human factors research on data modeling: A review of prior research, an extended framework and future research directions*. <http://services.igi-global.com/resolvedoi/resolve.aspx?>
- Tsichritzis, D. C., & Lochovsky, F. H. (1982). *Data models*. Englewood Cliffs, N.J.: Prentice-Hall.
- Tversky, A., & Kahneman, D. (1974). Judgment under uncertainty: Heuristics and biases. *Science*, 185(4157), 1124–1131.
- Villedieu, J. (2015). How the ICIJ used linkurious to reveal the secrets hidden in the swiss leaks data. *Linkurious Blog*. (<https://linkurio.us/blog/author/jean/>).
- Wand, Y., Monarchi, D. E., Parsons, J., & Woo, C. C. (1995). Theoretical foundations for conceptual modeling in information systems development. *Decision Support Systems*, 15(4), 285–304.
- Wand, Y., & Weber, R. (1990). Mario Bunge’s ontology as a formal foundation for information systems concepts. In P. Weingartner & G. Dorn (eds.), *Studies on Mario Bunge’s Treatise* (pp. 123–150). Atlanta, GA : Rodopi.
- Wand, Y., & Weber, R. (2002). Research commentary: Information systems and conceptual modeling - a research agenda. *Information Systems Research*, 13(4), 363–376.
- Wand, Y., & Weber, R. (2006). On ontological foundations of conceptual modeling: A response to Wyssusek. *Scandinavian Journal of Information Systems*, 18(1), 1–11.
- Wattal, S., Schuff, D., Mandviwalla, M., & Williams, C. B. (2010). Web 2.0 and politics: The 2008 U.S. presidential election and an e-politics research agenda. *MIS Quarterly*, 34(4), 669–688.
- Wiggins, A., Bonney, R., Graham, E., Henderson, S., Kelling, S., LeBuhn, G., ... Newman, G. (2013). *Data Management Guide for Public Participation in Scientific Research*. 1–41. <https://www.dataone.org/sites/all/documents/DataONE-PPSR-DataManagementGuide.pdf>
- Wyssusek, B. (2006). On ontological foundations of conceptual modelling. *Scandinavian Journal of Information Systems*, 18(1), 63–80.
- Yogev, S., Roitman, H., Carmel, D., & Zwerdling, N. (2012). *Towards expressive exploratory search over entity-relationship data*, Presented at the Proceedings of the 21st international conference companion on World Wide Web, ACM, pp. 83–92.