Which Model Does Not Belong

A Dialogue

Michael R. Gryk
School of Information Sciences
University of Illinois, Urbana-Champaign
Champaign, Illinois USA
gryk2@illinois.edu

Bertram Ludäscher
School of Information Sciences
University of Illinois, Urbana-Champaign
Champaign, Illinois USA
ludaesch@illinois.edu

ABSTRACT

Conceptual models can serve multiple purposes: communication of information between stakeholders, information abstraction and generalization, and information organization for archival and retrieval. An ongoing research question is how to formally define the fit-for-purpose of a conceptual model as well as to define metrics or tests to determine whether a given model faithfully supports a designated purpose.

This paper summarizes preliminary investigations in this area by presenting toy problems along with different conceptual models for the system under study. It is argued that the different models are adequate in supporting a sophisticated query and yet they adopt different normalization schemes and will differ in expressiveness depending on the implied purpose of the models. As the subtitle suggests, this work is intended to be primarily exploratory as to the constraints a formal system would require in defining the "usefulness", "expressiveness" and "equivalence" of conceptual models.

CCS CONCEPTS

• Database Design and Models • Query Languages

KEYWORDS

Conceptual Models, Relational Queries, Datalog, Taxonomies

1 Introduction

A monk was summoned to the Buddha's chamber. Upon a table were four pots: three gold and one silver. The first gold pot was large with ornate handles. The second could have been its younger sibling, sharing the same shape and handles yet of a diminutive size. The third was as big as the first but had no handles. The silver pot was large with handles.

B: Tell me my student, which of these pots is unlike the others?

M: The silver one of course. The others are made of gold.

B: Please hoist each one above your head.

M: Ah, master. I beg forgiveness. The one without handles is truly unique.

B: Would you please use them to milk the cow.

M: Once again master, I have changed my mind. The smaller one is inferior.

B: Several times I have given you a task and each time you have made a different choice. What task could I assign which would convince you to choose the large, handled pot of gold?

M: There is no such task, master. That pot is not unique in any way.

B: That is what makes it truly unique.

2 B's Story

The koan in the introduction contains what is referred to as a "Which One Doesn't Belong" (WODB) problem (Danielson, 2016). This type of problem typically challenges one to identify the object which is not like the others. In educational contexts (e.g., Sesame Street, K-12) the puzzle is sometimes not primarily about finding a single "right" answer, but the fact that there might be several answers, and the point becomes to articulate a *justification* for the chosen answer. A feature in the koan is that the chosen object (the first pot) differs from all others via a "meta-property": It is the only object *without* a unique property. In this section we introduce a relational model that can be used to systematically analyze WODB problems and that also sheds some light on the underlying conceptual modeling issues.

Figure 1: Two simple WODB problems: Examples 1 and 2.

The four objects `o1`, `o2`, `o3`, and `o4` depicted on the **left** in Figure 1 constitute a trivial WODB problem: the boxes differ only in a single property: their color. Clearly `o2` is not like the others, since it is the only object that is green. Another argument is that `o1`, `o3`, and `o4` are *indistinguishable*, i.e., we cannot single out any one of them without also "retrieving" the other two. Note that it is implicitly understood that the object-ID or the relative position is *not* part of the model: e.g., we can't say `o4` is special because it's the only red box located in the South-East quadrant of the puzzle.

Consider now the WODB puzzle on the **right** of Figure 1: Which object is not like the others? In this second example, o1 and o4 are indistinguishable and we might be tempted to answer either o2 ("it's the only blue box") or o3 ("it's the only blue circle"). To resolve the ambiguity, we can employ a relational database *D* and devise a formal justification using a query *Q* that picks the object we deem to be unlike all others. *D* consists of a set of facts prop(*X,P,V*) stating that object *X* has a property *P* and value *V*:

```
prop(o1,shape,box).  prop(o1,color,red).  prop(o1,size,large).
prop(o2,shape,box).  prop(o2,color,blue). prop(o2,size,large).
prop(o3,shape,circle). prop(o3,color,blue). prop(o3,size,large).
prop(o4,shape,box).  prop(o4,color,red).  prop(o4,size,large).
```

For *Q* we choose: "Object *X* is unlike the others if *X* has a property *P* and value *V* that no other object has". In Datalog, we specify:

```
unique(X,P,V) :- prop(X,P,V), not another(X,P,V).        (1)
another(X,P,V) :- prop(X,P,V), prop(X2,P,V), X != X2.    (2)
```

Rule (1) says that X is unique w.r.t. property P and value V if X is the *only* object with property P and value V. The auxiliary rule (2) finds objects X with property P and value V for which another object X2 exists having the same property/value pair as X. Therefore if another(X,P,V) holds, it means that X is *not* unique w.r.t. P and V. Evaluating query *Q* on database *D* yields a unique answer *A* = *Q*(*D*) for the right puzzle in Figure 1:

```
unique(o3,shape,circle)
```

Why is the object o2 not among the answers here? After all, o2 is the only blue box! However, we have modeled color and shape as independent properties: o2 is a box (and there are other boxes), and o2 is blue (and there are other blue objects). In contrast, o3 is the only object having value circle for shape. The same rules (1) and (2) can also be applied to the database instance *D* for Example 1 on the **left** of Figure 1. In that case, we get the expected answer:

```
unique(o2,color,green)
```

Now consider the slightly more challenging examples in Figure 2: Example 3 on the **left** in Figure 2, yields the following answer when running *Q*, i.e., rules (1) and (2), on the model database *D*:

```
unique(o1,size,small)
unique(o3,shape,circle)
```

This means that two objects are now "equally special": o1 because it has the unique property of being small, and o3 (as in Example 2) because it is the only circle.

An interesting aspect of these "equally special" solutions is that a *minimal change* will turn the justification for selecting o1 into a justification for o3 and vice versa! The property small is to o1 what the property circle is to o3. Formally, the justification (i.e., derivation in terms of a Datalog proof tree) of the first solution can be changed into one for the second solution (and vice versa), simply by swapping the properties small and circle.

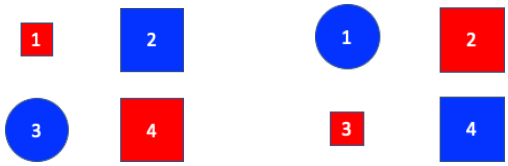

Figure 2: Two ambiguous WODB puzzles: Examples 3 and 4.

This last aspect seems to depend on the choice of the query *Q* to select the object that doesn't belong. Interestingly, there are aspects of the model that are *independent* of any choice for *Q*, as can be seen from the following argument: Consider any model M1 of the problem on the left in Figure 2: it associates with each object exactly three properties (for shape, color, and size). If we swap the properties small ⇔ circle, blue ⇔ red, and large ⇔ box, we obtain a new model M2 that describes the WODB situation on the **right** of Figure 2. Interestingly, the two models are *isomorphic* under this permutation of domain elements, *mutatis mutandis*: see Figure 3 and compare with Example 4 on the right in Figure 2. With IDs swapped (o1 ⇔ o3 and o2 ⇔ o4), Example 3 becomes Example 4!

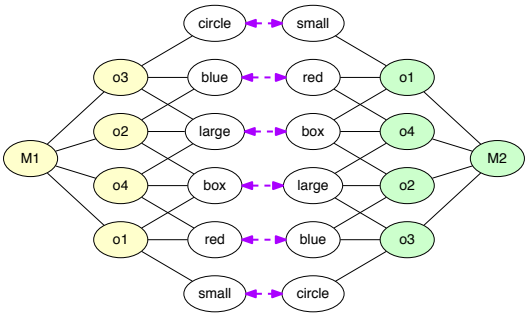

Figure 3: Models M1 and M2 are isomorphic under the permutation small ⇔ circle, blue ⇔ red, and large ⇔ box.

This also means that for any query *Q* (e.g., the one used above), the answers will be isomorphic, too: e.g., the answer for Example 3 is unique(o1,size,small), unique(o3,shape,circle), and for Example 4 we have unique(o1,shape,circle), and unique(o3,size,small).

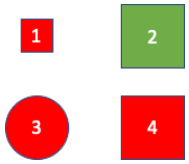

Figure 4: A confusing WODB puzzle: Example 5.

Last not least, consider Example 5 in Figure 4: Here, o1, o2, and o3 are each unique in a different way: unique(o1,size,small), unique(o2,color,green), unique(o3,shape,circle).

But this is odd: If *every* object is unique in its own way, *except* for one object that is "normal" (i.e., *not* unique in any way), then shouldn't that object be the one which doesn't belong!? Indeed, we can model such "meta" arguments formally:

```
special(O) :- unique(O,_,_).                    (3)
normal(O) :- prop(O,_,_), not special(O).       (4)
```

Here we just declare objects special that are unique w.r.t. some (unnamed) property and declare objects normal if they are not special. When running this query, we obtain a single normal object o4 (and three special objects o1, o2, and o3), and we can declare the single normal object to be the one which doesn't belong. Note that in this example, like in the previous example, each argument for one of the special objects o1, o2, and o3, can be turned into an argument for any other special object, simply by swapping the "chosen property". The way in which o4 is (meta-)special, in contrast, is different from the way o1, o2, and o3 are special. As in the opening koan, o4 stands out, since it is the only object that has no unique property.

3 M's Story

One of the central themes in last year's workshop paper (Gryk, 2019) was the assertion of George Box (1979) that "all models are wrong, some are useful." While this statement seems to be generally accepted by most scientists, the authors have noticed anecdotally that there are various degrees of specificity to which people believe the adage applies. Puzzled by this limited acceptance, MRG began playing a "modeling game" with his peers in which he would quickly transform a peer's conceptual model into something similar but different in order to highlight both subtle differences implicit in different modeling choices as well as to demonstrate there are often multiple equivalent models (as defined by a purpose.)

It was during this timeframe that BL presented his model and solution to the WODB problem during a weekly research meeting. What follows is an alternate model and solution to the problem.

3.1 M's Model

It is important to note that there is a "trick" involved with the meta-level of uniqueness contained in this WODB problem. That "trick" is that each property for an object has only two possible values. A pot can be big or small, gold or silver, handled or not. This binary nature of the properties is what allows for the identification of the unique-by-not-being unique object.[1]

With this is mind, one can easily transform the model from Section 2 into a model in which each object has a series of True/False attributes. For each Pot in our table, there is a binary attribute corresponding to the attributes gold, handled and large (Table 1).

Table 1. Alternate model for the Which One Doesn't Belong Problem.

POT	IS GOLD	HAS HANDLES	IS LARGE
1	X	X	X
2	X	X	
3	X		X
4		X	X

The unique pot is very easy to spot with this conceptual model; it is the one in which all attributes are True.

$$\sigma_{isGold,hasHandles,isLarge}(\text{Table 1}) \quad (5)$$

This new model succeeds by changing the reification of the properties of the object from values in table to named attributes. For a simple toy problem, this model is an acceptable alternative to the one in Section 2. However it could be argued that such a model will not scale. What if the WODB problem was extended to support objects with dozens or hundreds of attributes?

Yet this model is both useful and used by an entire domain of scientists. One early iteration of the toy problem used animals with various properties: Predators vs. Prey, Stripes vs Uniform Coats, etc. This version of the problem had a zebra and a tiger as examples and was an obvious example of the character matrix used by taxonomists (as shown in Table 2.)

Table 2. Example of a Character-taxon matrix. Adapted from Thomer, *et al.*, 2018.

TAXON	HAS 5 FINGERS	HAS FUR	LAYS EGGS
LION	X	X	
LIZARD	X		X
PLATYPUS	X	X	X
ZEBRA		X	

Character-taxon matrices as shown above are used to identify significant traits of organisms in order to help build a phylogenetic or evolutionary tree of life. Building such trees assumes that organisms with similar traits are more closely related than ones with dissimilar traits. While a simple concept, the use of such matrices requires much care (Vermeij, 1999).

The similarity between the WODB model (Table 1) and the character-taxon matrix (Table 2) is undeniable. Yet, there is still a question as to whether these two models are equivalent, regardless of the obvious similarity. In the case of character matrices, the similarity of traits are used to define evolutionary relationships. For instance, one might draw the conclusion from Table 2 that the platypus is the common ancestor to lions, lizards and zebra[2]. Is that

[1] The importance of the binary properties is left as an exercise for the reader. Would it be possible to spin this yarn if there were silver, gold and pewter pots?

[2] Mammals, marsupials and lizards are believed to have a common amniote ancestor.

conclusion a defined purpose of the model? Or are we reading into the model something which it was not intended to convey.

Similarly, one might assume that all of the pots in the toy problem were derived from the first pot. While it is left as a rhetorical question whether this is a defined role of the model or not, the first pot *was* actually the common ancestor from which the other ones were derived.

4 Conclusions

"All models are wrong, some are useful." Wrong is a characterization regarding some truth or gold standard of correctness and Box's adage warns us not to believe any model can express truth. Yet if some models are useful (and alternatively others are not), how can usefulness be defined, measured and validated?

The authors present no answer to this question but hope to begin a discussion, a dialogue to which others are invited. Two models have been presented in the context of a toy problem. Of course both models are wrong; are either of them useful? Can their usefulness be defined? In the case of B's model the usefulness appears to be linked to the query language which provides a justification for the uniqueness of the "normal" item. In the case of M's model the usefulness appears to be limited by the requirement that all properties are binary. Yet, that modeling paradigm is ubiquitous in biological taxonomy classification.

The title of the paper is "Which Model Does Not Belong" which poses the last unanswered question. Assuming we can define the usefulness of the two models, is there a method for probing their equivalence? Is one of these two models more useful than the other or is it impossible to identify which model does not belong?

ACKNOWLEDGMENTS

The authors thank Sahil Gupta who has created analysis tools for studying WODB problems, using his Possible Worlds Explorer tool (Gupta, 2020). In particular, his latest tool can identify the automorphisms of WODB instances.[3]

REFERENCES

George Box, 1979. Robustness in the Strategy of Scientific Model Building. New York.

Christopher Danielson, 2016. Which One Doesn't Belong? Playing with Shapes.

Michael R. Gryk, 2019. Matryoshka Modeling: Building one conceptual model within another. 2019 JCDL Workshop on Conceptual Modeling.

Andrea K. Thomer, Michael B. Twidale, and Matthew J. Yoder. 2018. Transforming Taxonomic Interfaces: "Arm's Length" Cooperative Work and the Maintenance of a Long-lived Classification System. Proc. ACM Hum.-Comput. Interact. 2, CSCW, Article 173 (November 2018), 23 pages. DOI:https://doi.org/10.1145/3274442

Geerat J. Vermeij, 1999. A Serious Matter with Character-taxon Matrices. Paleobiology, 25(4), 431-3.

Gupta Sahil, 2020. Possible Worlds Explorer: Combining Declarative Programming with Jupyter Notebooks, MS thesis, University of Illinois at Urbana-Champaign. Open source code: https://github.com/idaks/PW-explorer.

[3] https://github.com/idaks/WODB