Data Design

06.01.2018 Arve Meisingset

Purpose

The purpose of this paper is to illustrate how data design deviates from ontology, and to give references to some of my earlier writings on the topic.

Introduction

In my book, Formal Description of Anything in the Universe, sections 1-2, we show how to analyse what exists in a Universe of Discourse (UoD). In section 3, we show how data may denote the phenomena in the UoD, and show that some data may not denote anything.

In section 12 of the book, we discuss design of IT systems, and identify the challenges with identifying the right UoD.

Section 16 of the book shows how to state Denotations. Colloquial language does not make a proper distinction between inscription, prescription and description. They call everything for a description, and therefore wrongly think that each term describes something, which they quite often do not.

Section 19 of the book discusses classes and derivations. Both phenomena and data need classes. Derivations may take place among phenomena, or take place between data without having corresponding phenomena.

ITU-T Recommendation Z.352 contains a set of guidelines for data design.

In addition to the above, I have written a paper explaining that the phenomena are themselves data inside some observer; they are not entities out in some reality. An ontology is about the phenomena. A meta-ontology is identical to the language for defining data.

This short paper discusses data design through use of a very small example.

Data design example

Suppose you want to manage People. Is People the right class, or do you mean Human, Household, Legal person, Car owner, Inhabitant or other? Your choice will have great impact on the scope, contents and capability of your IT system.

You may want to manage the People within a Country. If you mean the People of the Country, you may want to manage its contained Inhabitant-s, and not Person-s. The choice of the class Inhabitant follows from the choice of the class Country. Inhabitant-s do not comprise every Person who currently stays in the Country. An Inhabitant of the Country may even stay outside the Country.

If we manage Inhabitant-s of one Country only, the database may only store records of the class Inhabitant and not Country. At the user interface, we may want to state that the Inhabitant-s belong to a specific Country. Hence, at the user interface we need both the class Inhabitant and the class Country, even when there is only one Country instance.

From this example, we see that the data structures of the database and of the user interface are not identical. Naming conventions, and many other details, will be different, as well.

Most IT professionals are only defining the data structures of the databases, and do not understand that we need many more data structures, and mappings between them. The most prominent data structure of any IT system is the data structure of the user interface. This data structure we call the External terminology schema. This schema defines the complete terminology and grammar of the user interface. See the book, section 10.

On ontology

Do Country-s exist in the UoD? They may not. A Country is an agreement between people within or outside the Country. Any agreement is a set of data, and is not a phenomenon (within a UoD).

If we try to observe the People and their behaviour, we may not be able to identify Country-s or their Inhabitant-s. It is only by data, eg. the passports, that the Country-s are made distinguishable.

Introduction of extra notions, like Country, that help us to overview or get insight into subordinate objects is typical for work of data design. These extra notions may even be used as name spaces for the subordinate objects.

The data class Country makes us define the data class Inhabitant.

A Country is a subject, from which and for which, we are registering Inhabitant-s. A Country is not an object to be observed.

The Inhabitant-s are the objects that we observe and register. These objects we choose to identify by a Civil registration number within the scope of a Country. Neither Inhabitant-s nor Civil registration number-s exist out in the UoD.

Do People exist in the UoD? Maybe, but they are likely not observed from Country. Are they registered by Name and Address, ie. relative to a Geographical location? Or are People only observed from other People, ie. my father, my son, my friend etc? If so, one Person may be a separate Father for each Child, etc. Hence, Father may not be a Person.

If each Inhabitant is local to a Country, there may be some Person-s that are Inhabitant-s in several Country-s. Hence, there may be several Inhabitant-s, each having different values, for each Person.

We may cross-reference the Inhabitants. Then we get the class Cross-referenced inhabitant, which may have its own identifier and attributes, different from Inhabitant. Each Cross-referenced inhabitant is local to the Inhabitant that it is referring from. The Cross-referenced inhabitant may be dangling without referring to another Inhabitant. There is no phenomenon corresponding to a Cross-referenced inhabitant out in the UoD.

If a Person changes sex, then maybe the Civil registration number will have to be changed, as well. Hence, one Person may change Inhabitant in the same Country during his lifetime.

If a Person changes sex, his appearance and behaviour will change. Is he then the same Person?

We may not need to register the phenomena Person-s in our IT system, but we will need to define, design and manage data.

Each Person may have one Father. However, we may not know the Father of each Person. Hence, the rules for what must exist among the phenomena may not appear among the data – neither for Person-s nor for Inhabitant-s.

Under Country, we may add the attribute Number of inhabitants. This number is calculated by summarizing the registered Inhabitant-s in the database. This number, or something comparable to it, is not easily found in the UoD. The Number of inhabitants will not cover babies who are not yet registered, not deaths that are not yet registered, and not applications that are not fully processed. Also, a county may not yet have delivered its recent data, due to technical or administrative reasons; hence, previous data are used. This illustrates that we are talking about data, and not reality. Any interpreter of the data need to know this distinction.

We chose and design data that are efficient and flexible for management, give proper insight and overview.

Conclusion

This short paper illustrates how data may deviate from phenomena. Design of data depends on analysis of phenomena, but Data design is different from Ontology.