Ontology Meets Business - Applying Ontology to the Development of Business Information Systems

Matthew West

Shell International Petroleum Co. Ltd., Shell Centre, London, SE1 7NA, UK
Matthew.West@Shell.com and dr.matthew.west@gmail.com
http://www.matthew-west.org.uk

Abstract. Ontologies are often perceived as not useful for practical problems. This chapter shows that this is not true. We present an ontological framework to support development of business information systems with a focus on the conceptual data modeling phase. We introduce four-dimensional analysis, with spatio-temporal extents, and apply this approach to several examples from practical experience. We notice that four-dimensional analysis results in sets with unchanging membership, and show how some alternative set theoretic approaches have application in practice. We look more closely at properties, and in particular physical quantities. Finally, we look at how this affects the development of data models and give a case study of the development of Shell's Downstream Data Model.

1 Introduction

Ontology can seem remote from the "real world" in either its philosophical or computer science sense.

Philosophical ontology is the study of the sorts of things that exist, and is closely related to metaphysics. It is characterized by argumentation about the philosophical consistency of different ways of viewing the world. The computer science sense of ontology arises from the world of artificial intelligence, and is rooted in representations of the world in formal logical languages of various sorts. Neither of these have had any significant, if any, impact on the world of business.

On the other hand, the world of business is driven by data held in databases that are defined by the SQL language. Traditionally, the design of databases has been facilitated by the use of data models, using one of a variety of graphical notations, known as entity-relationship models, developed from the origin work of Chen [27].

The traditional approach to data modeling has been one of normalization. This is a bottom up approach to analysis that looks at the data, and seeks to minimize the holding of redundant (repeating) data. The elimination of different sorts of repeating patterns, by creating new entity types, gives rise to normalization to different levels. There are currently 6 levels of normalization recognized, and if you achieve all of these levels then you would say that your data model was in 6th Normal Form, or 6NF. The details of normalization are well documented in standard textbooks on data modeling such as [30]. What one notices doing analysis over a period is that fully normalized data models have a structure that reflects the objects of interest to the enterprise. This has lead to the development of data models directly using the objects of interest to the business, rather than analyzing the data bottom up [1]. However, these approaches have generally followed a naïve and intuitive approach to identifying the objects of interest, and whilst some efforts have been of high quality [28] the results have been far from uniform, with different practitioners applying different intuitions of varying quality to the analysis process.

Serious attempts to apply philosophical and computer science ontology to the development of data models can be found in [29] and [5]. This chapter builds on this work, explaining the practical application of ontologies to the development of information systems for business. In particular the relevance of ontology to data models, and reference data is considered.

The chapter brings together, revises, and extends work from a number of publications.

2 The Relationship Between Ontologies and Data Models

2.1 Ontology

An ontology is a theory of what exists, i.e. the sorts of things there are and the rules that govern them. An ontology can take many forms, have different levels of detail, define many or few rules, and be represented in many formalisms or none.

This definition varies from the one usually quoted for the computer science sense of ontology by Gruber [10] "An ontology is an explicit specification of a conceptualization." What Gruber means by "conceptualization" in this definition, is "an abstract, simplified view of the world that we want to represent." This is essentially what a theory is, but "conceptualization" has roots in conceptualism, where it is our ideas about the world that count, rather than the realist view that an ontology is directly about the real world. Finally, the use of "an explicit specification" means that the same theory represented in another form would be considered a different ontology, and here in particular, we are concerned with the same theories perhaps in different representations as we pass through the design process, but wishing to see the theories as what essentially represents the ontology, rather than a particular specification of the theory.

2.2 Data Model

A data model specifies the data we wish to hold about things of interest to the business using entity types, relationships, and attributes. This in turn may form the basis for the design of the database for one or more computer systems. The types of things of interest usually form the structuring basis for the selection of entity types, thus forming the link between data models and ontology. Indeed, a data model is one way of representing an ontology. However, some restrictions on the expressiveness of data models should be noted:

1. It is not possible to say that one entity type is an instance of another.
2. Individuals (broadly physical objects and activities) cannot be represented as entity types, because they do not have instances.
3. It is not possible to say that an instance of an entity type is a subtype of an entity type.

2.3 Ontology as Data Model or Reference Data

With a data modeling approach to ontology, a choice has to be made as to what parts of your ontology are represented as entity types, and which as reference data. The reference data in question here is data about the types of things of interest to the business. **Fig. 1** shows two ways in which it can be shown that P101 is a pump. In the first case P101 is an instance of the entity type pump. In the second case P101 is classified by the instance of equipment type pump.

Another restriction that can be seen here is that there is no way in the second case to show that pump is a subtype of the entity type equipment item. You can also see that the entity type, equipment type, is essentially part of the meta-model for the first case. This means this style of data model can have an unfamiliar feel to those with a background in ontology rather than data modeling. On the other hand, the ontology in the second case is easy to extend with other equipment types as reference data, even after the system has been built. The choice between these two approaches will depend on the purpose of the data model. Indeed, an important design choice when designing database systems, is where to place the divide in the ontology between what part is in the data model and what part is in the reference data.

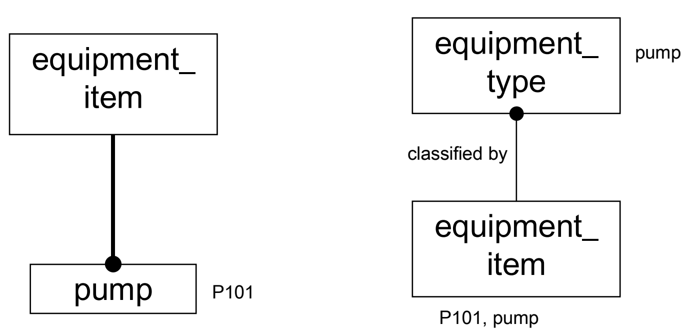

Fig. 1. Alternative ways to represent elements of an ontology in a data model or database.[1]

[1] Data models are drawn in this chapter using the EXPRESS notation defined in ISO 10303-11. The thin line is a one-to-many relationship with the "lollipop" at the one end. The thick line is a subtype/supertype relationship, with the subtype at the "lollipop" end.

3 Some Problems Found with Some Data Models

Key parts of any enterprise architecture are the data models of its information systems. These sometimes constrain the information that can be held because the ontology in the data model does not match the business reality. Ten common traps are identified in [1]. One of the examples, for relationship cardinalities that lose history, is presented here to illustrate the kind of problem that can be found.

3.1 Relationship Cardinalities that Lose History

Sometimes cardinalities are set to one-to-many, meaning one at a time, when the cardinalities are really many-to-many over time because the relationship is transferable.

3.1.1 Consequences

Imposing restrictions through the data structure means:

- Arbitrary or inappropriate restrictions are placed on the data that can be held.
- History data about a relationship cannot be held.
- The entity type will only work within the context defined. A change in business rules may require a change in the database structure.
- The resultant system is harder to share.

3.1.2 An Example - Ship

Fig. 2 shows that a Ship is registered at one Port and only one Port, under one name and only one name.

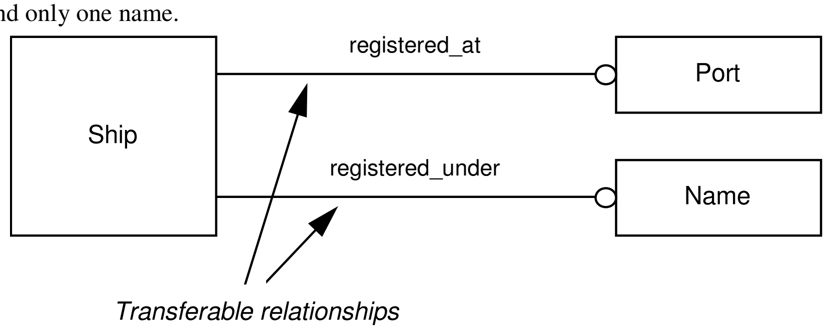

Fig. 2. Transferable relationships

However, what happens if you re-register a ship? How do you know what it was previously sailing as? The same applies to the name. If it changes you do not know that it refers to a vessel that you had blacklisted, or was an old friend.

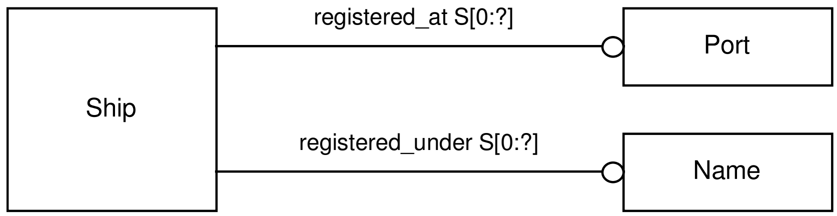

Fig. 3. Correct cardinalities for transferable relationships[2]

Fig. 3 shows the correct relationship cardinalities as many-to-many, which recognizes that a one-at-a-time relationship is potentially many-to-many over time. The problem was caused by modeling a business perspective, that we normally refer to a ship by its name and port of registration, rather than looking for what underlies that view.

Resolving the many-to-many relationships into entity types leads to a model as illustrated in **Fig. 4**.

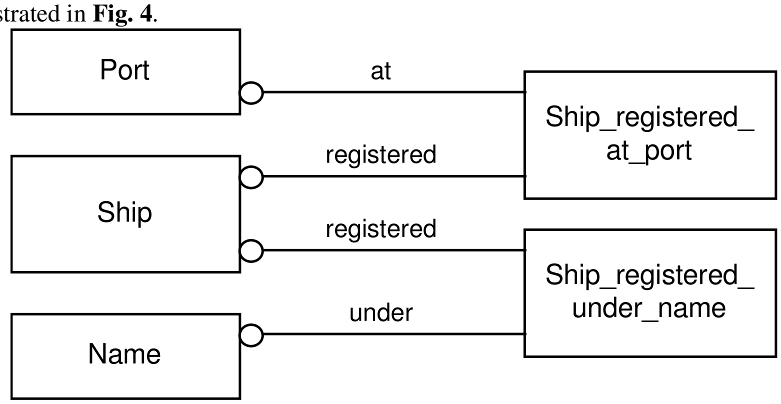

Fig. 4. Resolution of many-to-many relationships

However, in this case there is an activity that underlies both these relationships, Registration, and if this is recognized, then we can have one instead of two entity types representing the registration as shown in **Fig. 5** below.

[2] The S[0:?] means in this case that a Ship may be registered under a set of 0 or more Names, and registered at a set of 0 or more Ports, i.e. the relationships are many-to-many.

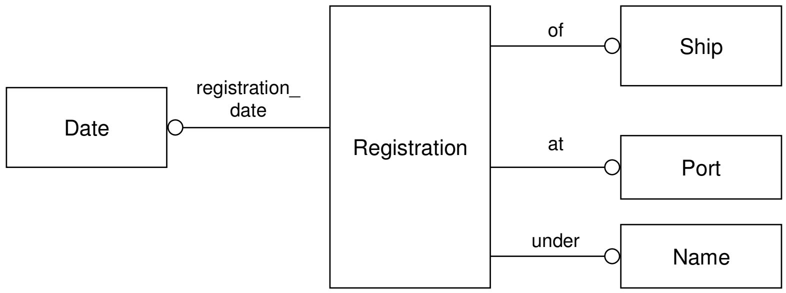

Fig. 5. Understanding that activities cause relationships

The relationships to the other entity types are one-to-many, and are now named in terms of the involvement of the entity type in the activity.

4 Some Principles for Data Modeling

Some general principles for avoiding traps, like the one illustrated above, were also identified in [1]. They are:

1. Entity types should represent, and be named after, the underlying nature of an object, not the role it plays in a particular context.
2. Entity types should be part of a subtype/supertype hierarchy (class hierarchy) in order to define a universal context for the model.
3. Activities and associations should be represented by entity types (not relationships or attributes).
4. Relationships (in the entity/relationship sense) cannot be referred to directly as objects, so should only be used to represent things that you do not need to refer to independently, such as the involvement of something in an activity or relationship.
5. Entity types should have a local identifier within a database or exchange file. These should be artificial and managed to be unique.
6. Candidate attributes should be suspected of representing relationships to other entity types.

The first of these principles makes clear the importance of "the nature of things" that is at the heart of ontology. A data model is an ontology, and as such makes ontological commitments, though these are rarely explicitly acknowledged.

5 A 4-Dimensionalist Ontological Framework

5.1 Introduction

Data model consistency is dependent on taking a common view of how to represent things across the business. Unfortunately there are many ways in which we can model the world. However, there are two main approaches, with on the whole minor variations, that dominate the philosophical literature. I will call these the 3 dimensional paradigm, and the 4 dimensional paradigm. The differences between these paradigms are illustrated in **Fig. 6** below.

The 3 dimensional paradigm says, for example, that all of me exists now, and that I pass through time, and therefore I do not have spatial parts. The 4-dimensional paradigm [2, 3] says that I am extended in time as well as space, and that I have temporal parts as well as spatial parts. An additional choice is whether identity of individuals is extensional. For a 3D approach it can be problematic to insist that only one object exists in one place at any time. However, under the 4 dimensional paradigm, since objects are extended in time as well as space, it is an option to take spatio-temporal extent as the basis for identity, and we do.

It may be noted that much of natural language seems to favor the 3 dimensional paradigm. I conjecture that perhaps this is because much of natural language is about the here and now, and so has become tuned to be efficient for that. However, it is perfectly possible to speak 4 dimensionally.

There is much philosophical debate around whether either or both of the paradigms are correct, however, this debate is beyond the scope of this chapter. Here the 4 dimensional paradigm is adopted, because it is seen as having change over time built in – rather than as something that is added on to the basic paradigm. This chapter then works through the consequences of applying this paradigm in practice in a number of areas.

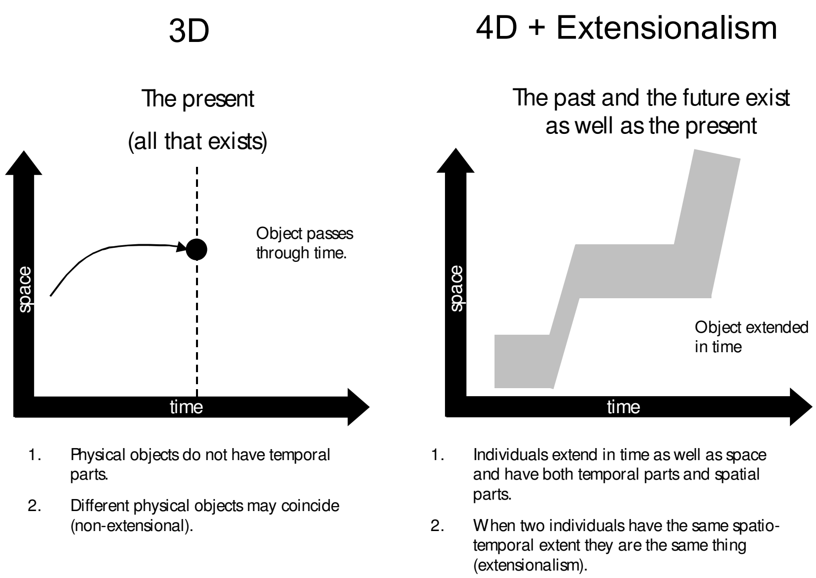

Fig. 6. 3-dimensionalism compared to 4-dimensionalism

It is conjectured that applying such ontological principles to data model development makes them more rigorous, and as a result, where data models are developed with the same ontological commitments, consistency is likely to be easier to achieve.

5.2 A 4-Dimensional Ontological Framework

We now present an ontological framework with a 4-dimensional foundation. It consists of:

- 4-dimensional spatio-temporal extents with extensional identity,
- Dissective and non-dissective classes,
- 4-Dimensional Patterns,
  - Ordinary physical objects,
  - Replaceable parts,
  - Intentionally constructed individuals,
  - Levels of reality for what things are constituted from,
  - Activities and events,
  - Roles as temporal parts of individuals,
  - Time,
  - Relationships as states with states of individuals as parts,
  - Possible Worlds for dealing with plans,
- Classes as sets, since membership does not change,

- Properties of various sorts including physical quantities.

Together these give the building blocks we need to model the world around us.

5.3 Spatio-temporal extents, Individuals and States

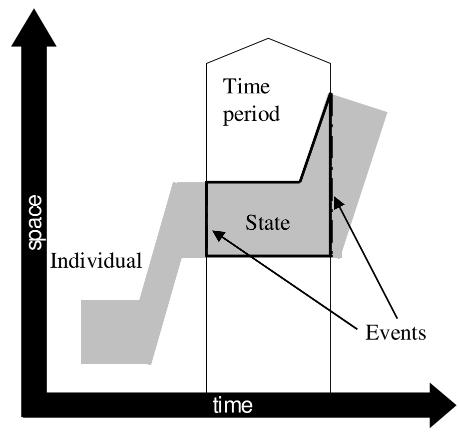

Fig. 7. Spatio-temporal extents, individuals, states, and events.

A spatio-temporal extent is any piece (not necessarily contiguous) of space-time; see **Fig. 7**. All possible spatio-temporal extents are allowed, which is not to say that you have to be interested in all of them.

Some spatio-temporal extents will be the whole lives of things that may be of interest such as persons, atoms, activities, and brooms. We call these individuals. Some spatio-temporal extents will be temporal parts of individuals, i.e. they will consist of the entire individual spatially for a part of its life. We call these states. It follows that an individual is a maximal state.

A state, whether or not it is also an individual, has a temporal boundary (strictly it may be spatio-temporal, but we will deal only with the simple case here) that marks its beginning and end. We call these events. An event is the change in state, not what brings about the change.

5.4 Time

One thing to notice in **Fig. 7** is how time appears. A point in time goes across all of space, and a period of time is a spatio-temporal extent across all space, bounded by two points in time.

A 4D treatment of the relationships between objects, activities, and time is presented in [4]. **Fig. 8** shows some key relationships objects have to time. Note that the black pieces are in fact one object that has periods of non-existence, as happens sometimes.

- Historical closure: the spatio-temporal extent that is all space (i.e. everything going on at the same time) whilst the object exists.
- Pre-history: the spatio-temporal extent that is all space before the first point in time when the object existed.
- Post-history: the spatio-temporal extent that is all space after the point in time when the object has finally ceased to exist.
- Extended history: the spatio-temporal extent that is all space from the point in time that the object first existed, to the point in time after which it no longer ever exists.
- Extended pre-history: the spatio-temporal extent that is all space until the last point in time that the object exists.
- Extended post-history: the spatio-temporal extent that is all space after the first point in time that the object existed.

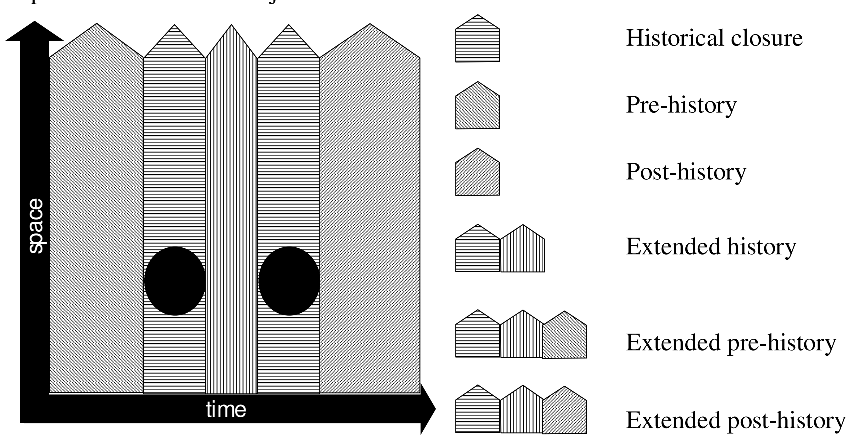

Fig. 8. The relationship between an object and time

Other useful operators are also defined:

- Historical connection: When the historical closure of two objects meet or overlap.
- Historical part: when the part falls within the historical closure of the whole.

This at least is the case for pure time and pure periods. However, in business we might be interested in something slightly different. Imagine that you are running a business that works in lots of different time-zones around the world, and that you need to keep global accounts of your sales on a daily basis. What then is your global day? **Fig. 9** below illustrates how this looks as a spatio-temporal extent. Of course this is still relatively simple:

- There is summer time,
- Not all offsets are one hour,
- The start and end of the business day might be at different times (and not say midnight).

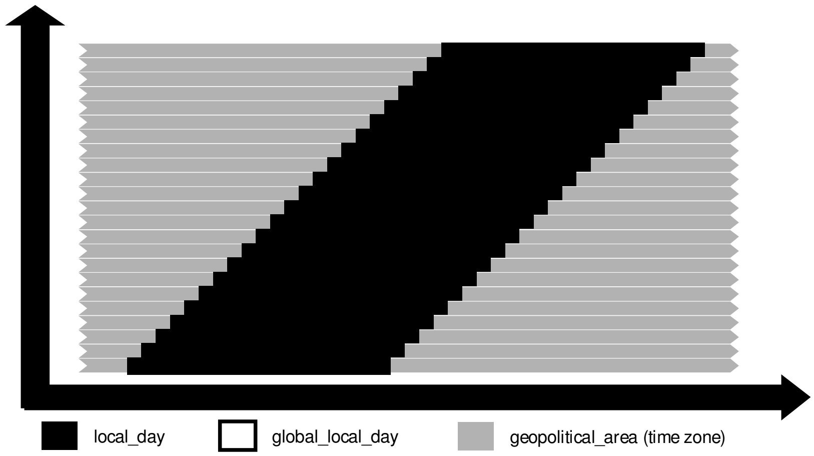

Fig. 9. Local, and global local days

However, space-time maps, and the 4D paradigm enable you to show these things in an intuitive way.

5.5 Different Sorts of Physical Objects

One of the consequences of a 4 dimensional approach is that you can examine the space-time patterns that different sorts of individuals exhibit. Indeed, you can use the spatio-temporal patterns that individuals can exhibit to identify the sorts of things there are.

5.5.1 Ordinary Physical Objects

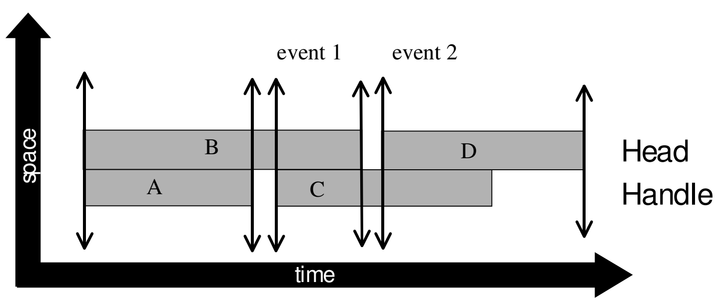

Fig. 10. A broom, with various heads and handles.

Ordinary physical objects, like the broom illustrated in **Fig. 10**, are expected to have continuous existence[3], but they can change parts or lose parts over time without losing identity, but they cannot change all their parts at the same time. Thus, whilst at the end of the period shown in **Fig. 10**, none of the original parts of the broom are now part of the broom, we say it is the same broom because not all the parts changed at once. Organisms are similar in renewing themselves by gaining and losing parts.

5.5.2 Intentional Objects

Many of the objects in the world around us only exist because we are here: money, companies, agreements etc. Here we look at how these exist and what implications this has. We call such objects intentional. The view I have taken in this case follows the work of John Searle [12].

Functional Objects

The simplest types of intentional object are functional object, this is where a function is imposed on an object, and has that function because of that assignment by someone. So a stone becomes a paperweight because someone (anyone) said so, and whilst they said so. This principle also applies to most types of equipment that man (and some animals) use for various purposes such as pumps and screwdrivers. Note that these objects consist of natural materials, and the function is put on them over and above being an amount of these natural materials.

Socially Constructed Objects

Socially constructed objects require the agreement of at least two people to exist: contracts, companies, money, etc. For example, money requires not only the authority of the issuer, but the acceptance of the populace who use it as money. A key aspect of socially constructed objects is that they need a process to manage their life, since they do not come into existence except by human will.

5.5.3 Levels of Reality

One challenging problem that 4-dimensionalism can help to explain, is the apparent coincidence of different objects. A simple example is presented in **Fig. 11** below.

The problem arises when, because of the way that parts are arranged, the whole has emergent properties that are not present in the simple aggregate of the parts. So in this case:

- When the nut and bolt are screwed together, they act as a fastener,
- When the steel they consist of is formed into the shape required, they have the properties of a nut and bolt respectively,
- When iron and carbon are mixed and arranged appropriately, they make steel,

---

[3] This can be questioned. For example, when a car or watch is disassembled for repair or maintenance, is it still a car, or does the absence of emergent properties mean it is just a collection of parts?

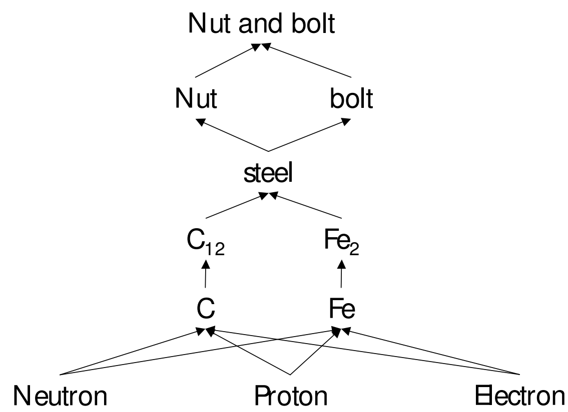

Fig. 11. Some levels of reality in a nut and bolt

- Carbon and iron molecules are arrangements of carbon and iron atoms respectively,
- A carbon or iron atom is an arrangement of particular numbers of protons, neutrons and electrons.

Now at a point in time, each of these is coincident. However, when you look over time, you can see that the spatio-temporal extents for each level are different, as illustrated in Fig. 12 below.

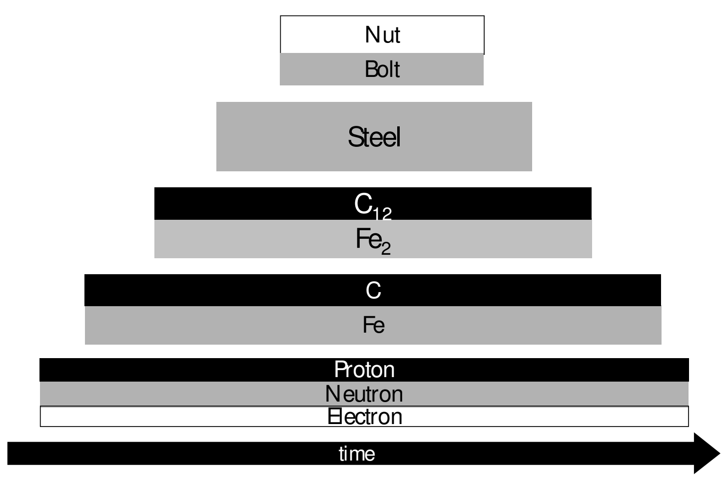

Fig. 12. The levels as potentially distinct objects

A problem case that is often used in the literature, is of the two pieces of clay that are brought together to form a vase, which at some later time is broken into a number of pieces, see **Fig. 13** below.

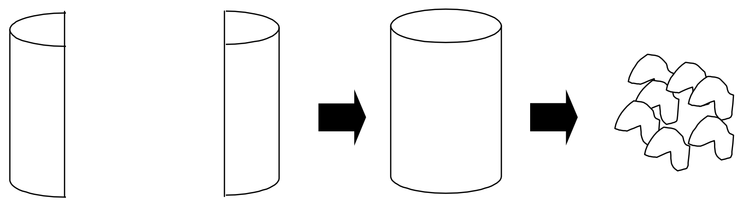

Fig. 13. The example of the coincident piece of clay and vase

Neither the clay nor the pot exists as one piece before the two parts are brought together or after the vase is smashed. So there is one object that is both the vase and the piece of clay it is made from. The consequence of this for an extensionalist is that the same individual can belong to classes at different levels of reality. This in turn means that a level of reality is in the end about the relationship between the classes, rather than between particular individuals, where the rule is that it must be possible for individuals to have a different spatio-temporal extent at the different levels, rather than that they must be different.

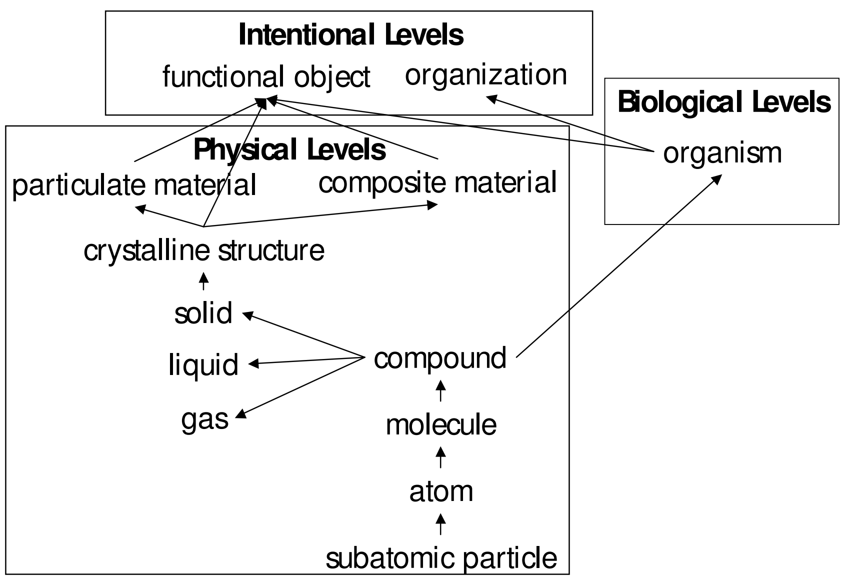

Fig. 14. Levels of reality implicit in ISO 15926-2

Fig. 14 above shows the levels of reality that are implicit in the data model of ISO 15926-2 [5]. It should be noted that, because any whole that has emergent properties because of the arrangement of its parts is a new level, many of these levels may be strata or sub-strata rather than distinct levels, and are certainly not complete. However, the levels in the physical strata are thought to be distinct.

5.6 Plans and How Possible Worlds Can Support Them

A key issue in ontology is how to deal with what could be, as well as what is. We adopt an approach based on possible worlds [7]. This allows a number of things, including allowing worlds where the basic laws of physics might be different, and allowing alternative views of history or the future to be explored. However, in business, the practical use of this approach is for planning, where plans belong to a possible world, and the outcome belongs to the actual world, so that comparison can be made between them.

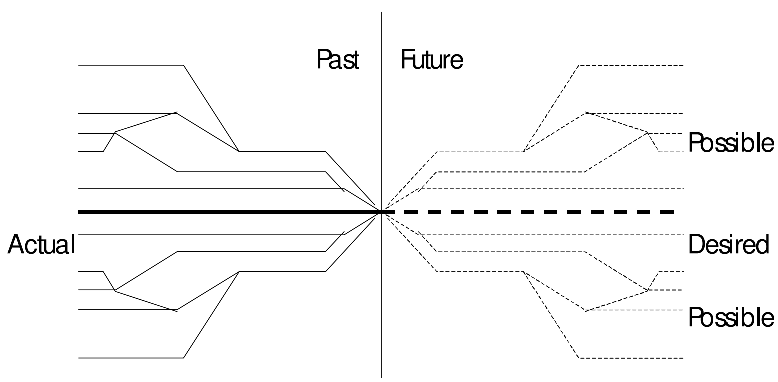

Fig. 15. Possible worlds

Fig. 15 illustrates how this can work. With a spatio-temporal approach to individuals, possible worlds can be allowed to intersect, with temporal parts of individuals being shared across possible worlds, since the possible world would be defined by its whole spatio-temporal extent, and only this would have to be unique.

5.6.1 Participation in Activities and Replaceable Parts

An activity is something that brings about change, i.e. causes an event (change of state). However, if an activity exists in space-time, it is not obvious what its spatio-temporal extent is. **Fig. 16** answers this question with an example of a game of football. It can be seen here that the football match consists of the states of the players (and other individuals) that participate in the game. This is true of all activities: that they consist of the participating states of their participants. Note that the parts may be scattered, both spatially and temporally.

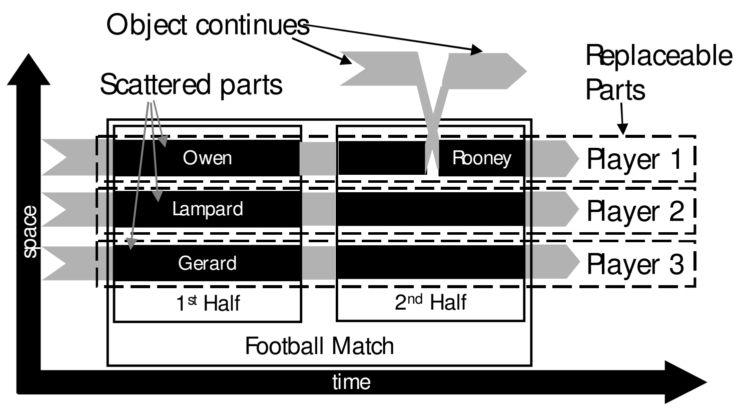

Fig. 16. A football match showing players as replaceable parts of a team, and of the football match itself

Fig. 16 can also be used to illustrate the principle of replaceable parts. A football team has 11 places. These are independent of both the number on the players back, and the role (goalkeeper, forward, defense) a player is allocated to. Further, one player may be substituted by another. Owen being substituted by Rooney in the middle of the second half of the game illustrates this in **Fig. 16**.

A key thing to be noted here is that all the parts of player 1 have been substituted at the same time, something that with ordinary physical objects would mean you had a new object. So this pattern of replaceable parts is distinctive.

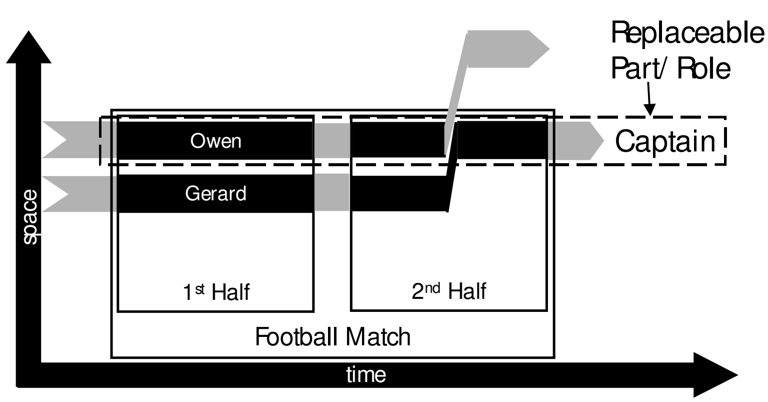

Fig. 17. Roles, like being team captain, are a type of replaceable part

Some replaceable parts can be roles. This is illustrated in **Fig.17**. This is another view of the same game. But here we see that Owen was captain whilst he was on the field, but when he left, the captaincy was given the Gerard. Notice that whilst Owen's numerical replacement was Rooney that did not mean that Rooney took over the role of captain. These can be seen as distinct, because they have different spatio-temporal extents. Other examples of replaceable parts that are roles are positions in an organization, such as President of the United States. A comforting thing about this analysis, compared to others that see roles as abstract objects, is that it makes sense to talk about shaking hands with (a state of) the president. There is only one object that you shake hands with, but it is a state of the President as well as say a state of Bill Clinton. Both are physical objects, they just have different patterns.

5.7 Roles of individuals in relationships

Historically many data models have taken a snapshot view of the world, which means that when change takes place, history is lost because it is overwritten. Recall the Ship example from earlier. In fact for relationships involving individuals, it may be the case that a relationship of a particular type holds for different objects at the same time, for different objects at different times, or even the same objects at different times.

One approach to dealing with this is to make the relationships temporal, we call such relationships associations. This is a 3D approach. However, from a 4D perspective some questions arise.

- If the relationships are temporal then they at least exist in time, so what is their spatio-temporal extent?
- If we can identify the spatio-temporal extent, what sort of thing are they?

The 4 dimensional approach is to manage change through recognising different states of individuals that are valid for a period of time, together with timeless relationships between these. This approach enables us to answer the questions above.

Two different patterns are presented for how different sorts of association can be represented in spatio-temporal terms. They are taken from [6], but revised and updated and including material from [31].

5.7.1 Relationships between two states of an individual

Fig. 18 below illustrates the case where an association represents a relationship between two states of individual things. To illustrate the model an example is given of how a wheel, "Wheel1", is part of a car, "Car1" from 1/1/2001 to 5/4/2001.

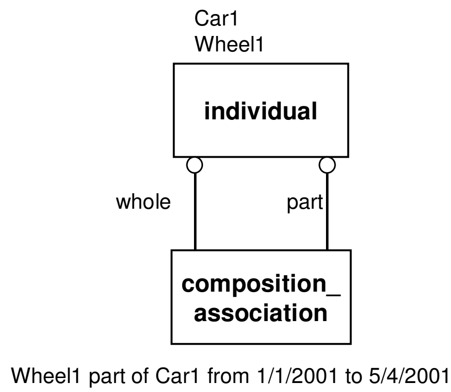

Fig. 18. Association between two individuals

Fig. 19 below shows this example as a space-time map, showing the different states of the car and wheel, as well as the whole life of the car and wheel.

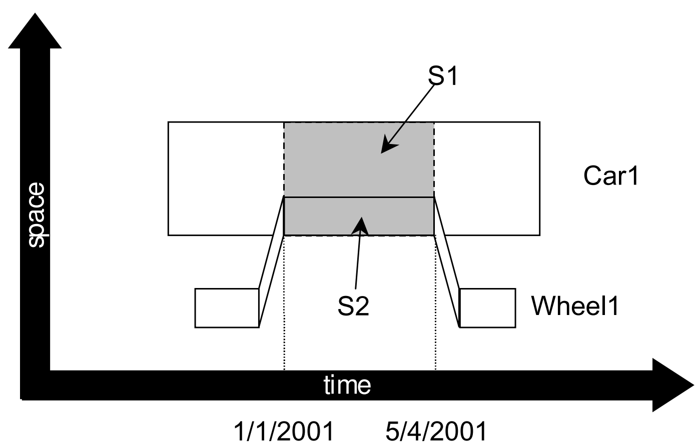

Fig. 19. A space-time map for composition

The diagram shows that in this case there is a state of Car1, S1, and a state of the Wheel1, S2, both with the same state and end date, and S2 is a part of S1. When this space-time map is modelled explicitly the result is found in **Fig. 20** below.

Here the states S1 and S2 are modelled explicitly. S1 is shown as being a temporal part of Car1, S2 is shown as being a temporal part of Wheel1, and S2 is shown as being a part of S1.

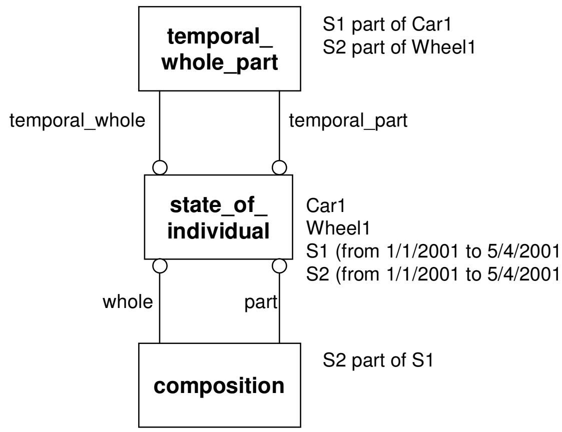

Fig. 20. Composition using states

Finally, although we have not shown it in the data models in order to keep them simple, subtypes of state_of_individual can be introduced for individual (whole life) and the roles played by various states, part, whole, temporal_part and temporal_whole, with the various objects being distributed appropriately.

5.7.2 Relationships between a state of an individual and a class

Fig. 21 shows a classification pattern. An example is given of where an individual, "Car1", is classified as being "Red" from 1/1/2001 to 4/3/2001.

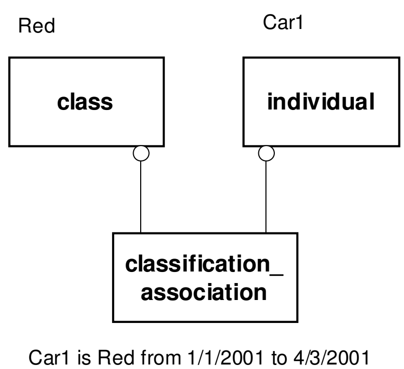

Fig. 21. An example of a classification association

If we examine what is happening here using a space-time map, **Fig. 22**, we see that there is a state of Car1 that is classified as being red (the shaded area).

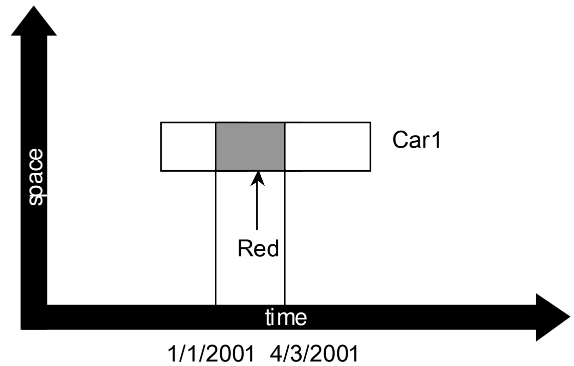

Fig. 22. A space-time map for classification of an individual

A data model that represents this space-time diagram is shown in **Fig. 23**. Here the state of the car that is red, state 1, is recognized, and timeless relationships are held to show:

1. That State 1 is a temporal part of Car 1, and
2. That State 1 is red.

Clearly, State 1 is always a part of Car 1, and State 1 is always red.

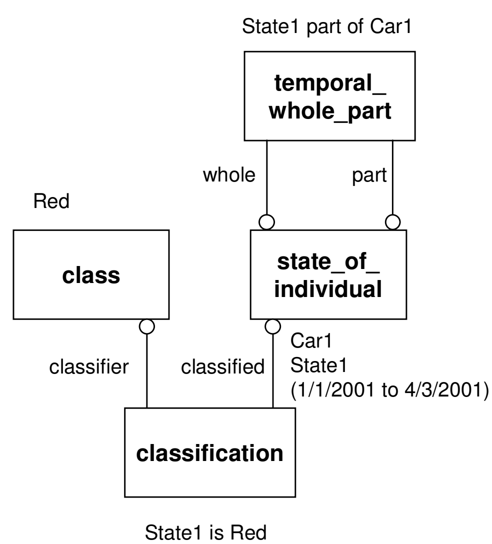

Fig. 23. Classification using states

5.8 Sets

This section looks at sets from the perspective of their use in ontology and data modeling, rather than the axioms that define sets. It is largely taken from an informative annex of [5].

5.8.1 What is a set?

A set is a thing that has members, and a set is defined by its membership (note: the null set is the set that has no members). That is, if two sets have the same members, they are the same set (so, for example, there is only one null set). If two sets have different members, they are different sets. In saying this, it is important to note that whilst its members define a set, it may be that at any point in time, not all the members of a set may be known.

5.8.2 Sets and 4-Dimensionalism

A problem with 3-dimensionalism is that because individuals do not have temporal parts (states), then the membership of sets changes over time. So, for example, taking the car that was red for a period of time, at one time the membership of the class Red Cars includes this car, and at another it does not. Now whilst the relationship between the class and the instance is of the same nature as set membership, the things that have the members are not strict sets.

An advantage of 4-dimensionalism is that because it is states of individuals that are instances of a set, the membership of the set is unchanging. Viewed from any point in time, that state of the car is red. This makes it much more straight forward to apply set theory. The only caveat being that we need to distinguish between sets where we know all the members, and sets where we do not. Thus under 4-dimensionalism classes are synonymous with sets.

5.8.3 Some different sorts of set theory

Single level sets

Single level sets allow sets to have members, but cannot themselves be members of sets. Entity relationship models where entity types cannot be members of other entity types are an example of single level sets. This is illustrated in **Fig. 24** below, where boxes indicate entity types, ellipses indicate instances, and arrows indicate which instances are members of which entity types.

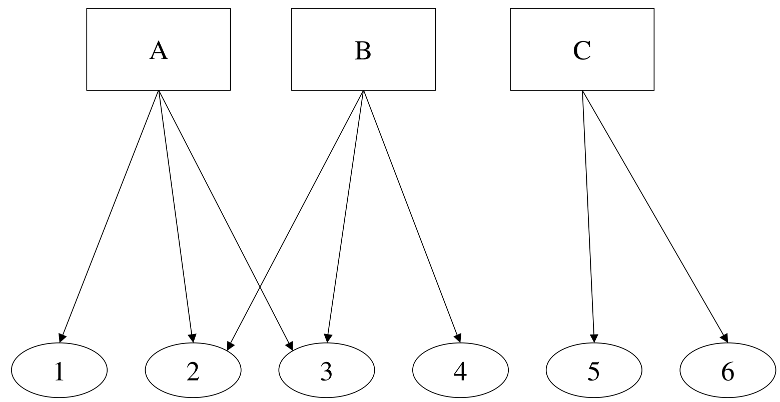

Fig. 24. Single level sets

In some cases, it is not allowed to be a member of more than one set.

Hierarchical sets

With hierarchical sets, sets at one level may be members of sets at the level above, but there is no crossing of levels. So sets can only have members in the level below. **Fig. 25** below illustrates this. Note that the relationship between levels is membership, and not specialization.

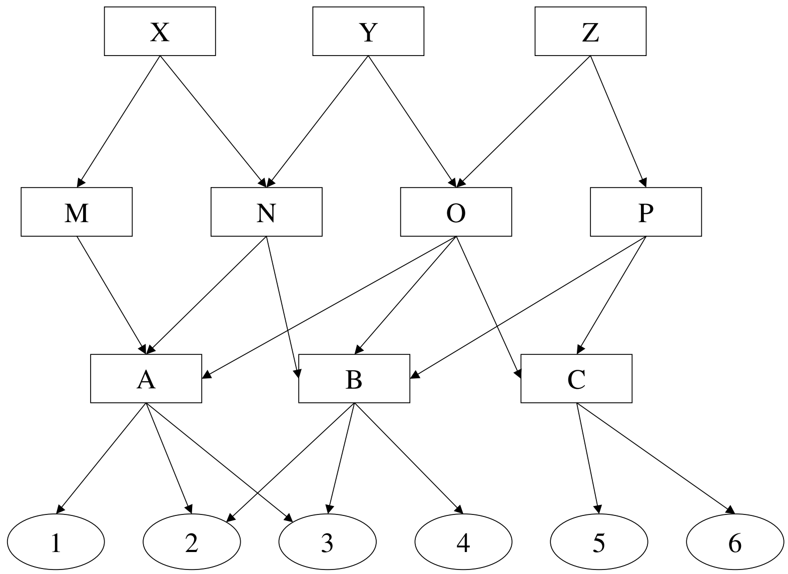

Fig. 25. An example of hierarchical sets

Hierarchical sets occur naturally and this is a useful pattern to look for (but not to force). It should be noted that hierarchical sets include single level sets as a subset. An example of hierarchical sets in use is in data model, meta-model, meta-meta-model approaches.

Another example is that of the powerset. A powerset is the set of all possible subsets of a set (including itself). An illustration of a powerset is given below in **Fig 26**. The representation is based on the Venn diagram, however, a set may be represented both as a set container, as an ellipse, and as an instance of a set, as a hexagon. A straight line links the two representations.

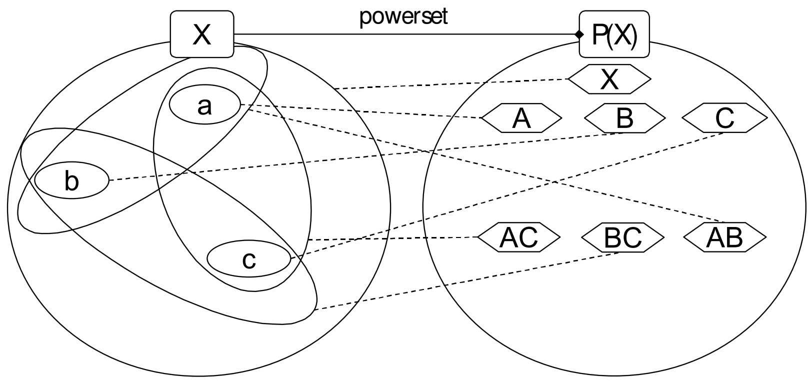

Fig. 26. An example of a powerset

In ISO 15926-2 [5] this pattern can be found, for example, in the relationship between the entity types possible_individual, class_of_individual, and class_of_class_of_individual, where class_of_individual is the powerset of possible_individual, and class_of_class_of_individual.

Well-founded sets

Well-founded sets are the sets of "standard" set theories such as Zermello-Fraenkel (ZF) set theory and von Neuman, Bernays, Goedel (VNBG) set theory that can be found in standard texts [8]. Well-founded sets can take members from any level below their own, but are not allowed membership loops (e.g. a set being a member of itself). This is illustrated in **Fig 27.** below.

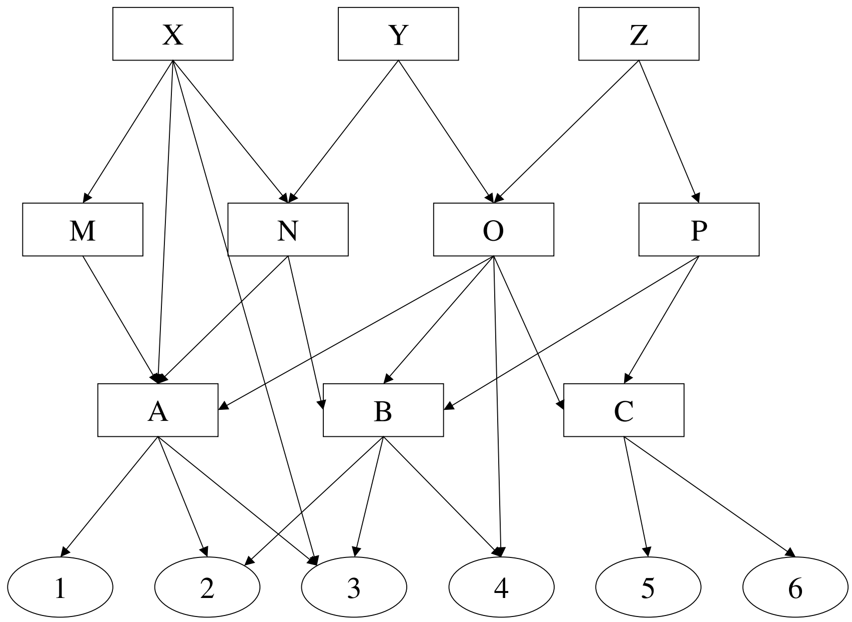

Fig. 27. An example of well-founded sets

This form of set theory was largely developed as a reaction (perhaps even an over-reaction) to Russell's Paradox. An early version of set theory developed by Frege allowed that for any predicate, there was a set that corresponded to that predicate. Russell gave an example of such a predicate that gave rise to a contradiction: the set of all sets that do not contain themselves. Either the resulting set is a member of itself (in which case it should not be) or it is not a member of itself (in which case it should be). Those working on set theory at the time felt that the best way to solve this problem was to disallow sets that had themselves as members (or other membership loops) and retain the property that any predicate (that did not involve a self reference or loop) would result in a set. However, this leaves some untidiness, for example, how does one say that a set is a set?[4]

It should be noted that well-founded sets include hierarchical sets as a subset.

Non-well-founded set theory

---

[4] One of the differences between the different versions of “standard” set theory is in how this question is answered.

The essence of non-well-founded sets (also known as hypersets) is to allow sets to be members of themselves, where the membership graphs can be constructed. This is illustrated in **Fig. 28** below.

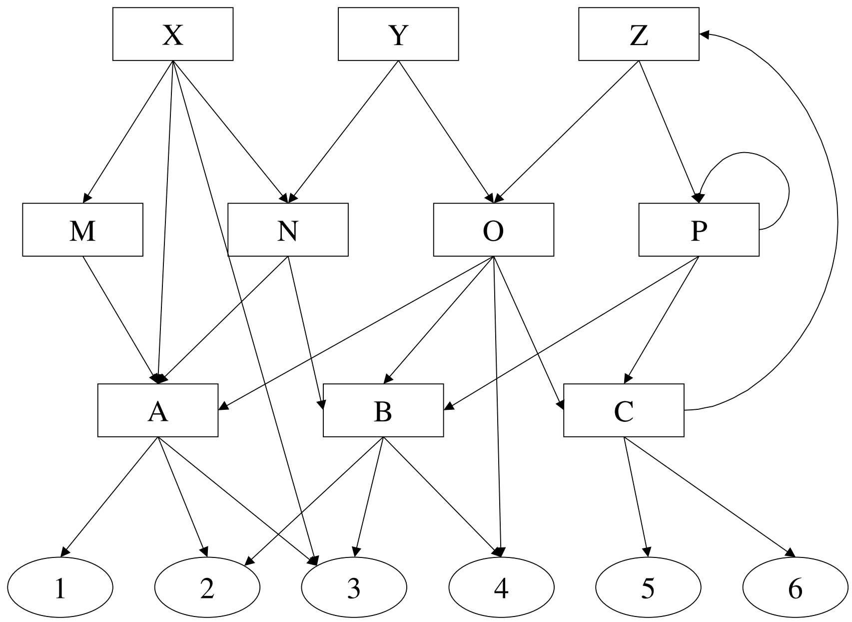

Fig. 28. An example of non-well-founded sets

In this case, Russell's Paradox is avoided by requiring that all sets can be constructed out of their members, so it is not assumed that there is a set that corresponds to any predicate. This allows useful things to be said that well-founded sets prevent, like "class is a class", "thing is a member of class", and "class is a member of thing".

It should be noted that non-well-founded sets include well-founded sets as a subset.

5.9 Dissective and Non-dissective Sets of Individuals

The ordinary sense of being dissective applies to the distinction between mass and count nouns. So for example, when you take a piece of water and divide in two you get two pieces of water, but when you take a person and divide them in two you do not get two people. One way of looking at this is to see it as a kind of inheritance, when a property of the whole is inherited by the part.

5.9.1 Temporally Dissective and Non-dissective Sets of Individuals

A particular sort of dissectiveness, relevant to 4-dimensionalism, is temporal dissectiveness. Here the question is whether a state (temporal part) of some spatio-temporal extent is also a member of a class, or has a relationship, that the whole has.

In general, states are useful precisely because they enable you to say something that is true at all times during which the state exists, so there is a presumption for dissectiveness. However, this only serves to make the exceptions more interesting. Obviously, temporal properties, such as the period that the state was a spatial part of, would not expect to be inherited. However, probably the most significant group of sets whose membership would not be inherited, are the subsets of individual, i.e. those spatio-temporal extents that are something for the whole of its life, e.g. car, person, flower. Clearly, a proper state of a car is not a car for the whole of its life. On the other hand, if you take the set, state of car, then a temporal part of a member is also a state of car. It is interesting to note that temporally non-dissective sets correspond closely to the traditional idea of natural kinds, and can perhaps be thought to usefully be the 4-dimensional definition of a natural kind.

5.10 Properties

On the other hand, the traditional idea of a property, with the notable exception of temporal properties, would seem to correspond closely to temporally dissective sets of individuals. Thus properties are inherited by states of the spatio-temporal extent to which they apply (although it is not only individuals that can have properties). **Fig. 29** below illustrates the structure of a property, using temperature as an example, using a modified form of Venn diagram.

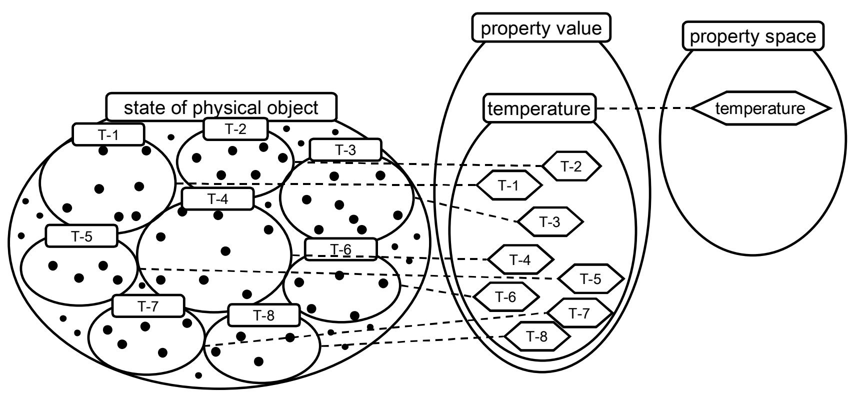

Fig. 29. Temperature as an example of property

The black dots in the left hand ellipse are members of the class, state of physical object. Those states that have a constant temperature are shown as members of the appropriate temperature property value, being a particular degree of hotness, such as 300 K. These are designated T-1 to T-8. The middle ellipse is the set of all property values, and T-1 to T-8 shown as hexagons, to show that here they are members rather than subsets of the class that contains them. They have a dashed line to link them to their representation as a set in the left hand ellipse. They are, of course, all shown as members of the temperature property space. Finally, the right hand ellipse shows that temperature is a member of the class property space. Notice that this is an example of hierarchical sets, as illustrated in **Fig. 25**.

Whilst this illustrates the common pattern shared by properties, there are a number of different types. Some of these are related to the structure of the property values, and others to the nature of the property type.

We can identify:

1. Intrinsic and extrinsic properties.
2. Direct and indirect properties, and
3. Unordered and ordered properties – including physical properties such as temperature,

5.10.1 Intrinsic and Extrinsic Properties

The distinction between intrinsic and extrinsic properties is a philosophical distinction, where an intrinsic property, say mass, depends only on the object whose mass is of interest, but the weight of that object depends on the objects relationship to the surrounding gravitational field.

5.10.2 Direct and Indirect Properties

The distinction between direct and indirect properties is somewhat pragmatic. Here a direct property is one that is expressed directly, like temperature. On the other hand, an indirect property is expressed in terms of some direct property. So for example, the Maximum Allowable Working Pressure (MAWP) of a boiler is expressed as a pressure, but it is not the observed pressure of the boiler. It is important not to think that an MAWP is a type of pressure, but recognize that it is in fact a property that makes reference to a pressure.

It is clear that all indirect properties are extrinsic, and that all intrinsic properties are direct, but it is not clear that all direct properties are intrinsic. Further, looking at a case like MAWP, it is clear that this property is a short cut for some more detailed analysis that explains the way the MAWP was derived, and the use that should be made of it. The world of engineering is full of such properties.

5.10.3 Unordered and Ordered Properties

A property is unordered when it makes no sense to talk of one value being greater than another. This can only happen when the values a property can take are discrete: an example might be a set of statuses something can take up.

Ordered properties typically map to an integer space (discrete properties) of to a real number space for continuous properties, though these refer to total ordering, and in principle partial ordering is also possible. For a totally ordered property there is at least one ordering function that orders the property values so that it can be determined for each pair of property values whether one is greater than the other, or whether they are the same property value.

5.10.4 Quantity Space

A quantity space is a class with structure, i.e. a property and a particular total ordering function. The most interesting of these are the physical quantities, such as temperature, pressure, volume, mass etc. Relatively little has been written in recent years about quantities, with Ellis [11] being one of the better writers.

5.10.5 Scales and Units of Measure

If we want to put a number against a property value, then we need a scale. A scale is a structure preserving isomorphic mapping between a quantity space and a number space, where the number space might be all or part of the real numbers or the integers.

Now a scale will have a unit of measure. Traditionally, this is taken to be the value of one on the scale, but I think it makes more sense to think of the unit of measure as a plus-one function for the scale.

5.10.6 Other Aspects of Physical Quantities

There is much more about physical quantities that has not been covered here. These include:

- Measurements of physical quantities,
- Measurement methods,
- Accuracy of measured values, and
- Dimensionality of physical quantities.

6 Impact of Applying Ontological Principles on Data Modeling

Conceptual modeling is an important activity in terms of both organizational understanding and systems development. Despite this importance, and evidence to suggest that ‘errors’ in modeling are increased by orders of magnitude later in the systems development and maintenance process, it is well noted that conceptual modeling remains more of an ‘art’ than science [14,15]. In addition, given an age of integration, it is increasingly recognized that semantic understanding and interoperability is a key challenge for organizations and their systems [16,17]. Semantic interoperability is a knowledge-level concept that provides the “ . . . ability to bridge semantic conflicts arising from differences in implicit meanings, perspectives, and assumptions, thus creating a semantically compatible information environment based on the agreed concepts between different business entities.” [17]. Ontology is an emerging mechanism for dealing with semantic interoperability.

An ontological framework, such as that described here, provides a basis for developing data models that are consistent across a wide range of information requirements and business processes. The benefits are:

- The same things get modeled the same way, irrespective of the context in which they arise because similar things turn up in the data model close to each other, so that differences can be examined to see if they are real or not.

- The treatment of change over time is built into the 4-dimensional modeling approach, rather than bolted on afterwards. This means that it is always present and is done in a consistent manner.
- Data models can be developed more quickly, more cheaply, and are of higher quality because there is less rework that arises from the consistent (re)use of the framework.

6.1 Case Study: Shell's Downstream Data Model

[13] reports on the development of Shell's Downstream Data Model (DDM).

6.1.1 Background

Shell is a global group of energy and petrochemicals companies. Shell Downstream encompasses all the activities necessary to transform crude oil into Shell petroleum products and petrochemicals, and deliver them around the world. Shell's Downstream Business refines, supplies, trades and ships crude oil worldwide, and manufactures, transports and markets fuels, lubricants, bitumen, LPG and bulk petrochemicals for domestic, transportation and industrial uses. Altogether, the organization employs some 80,000 people.

In an increasingly competitive downstream market, Shell assessed that the cost and complexity of business systems and processes provided an opportunity to improve performance. As a consequence, it has aggressively sought to achieve operational excellence through an ongoing program of global standardization. In practice the organization seeks to achieve such excellence primarily through a combination of (a) business portfolio improvements, (b) the introduction of global processes and standards underpinned by a simplified global organization and (c) the adoption of consistent behaviors to reinforce the perceived benefits of going global. Process streamlining forms a key component in the strategy to simplify and standardize the way that the organization does business, with the objectives of:

- Promoting more accurate and responsive customer interactions.
- Removing errors and rework.
- Reducing costs by eliminating 'noise' in business processes.
- Providing proven and simpler ways of doing things.

Unsurprisingly, the standardization of the critical IT systems is seen as key to the success of the streamlining initiative. Thus, a partner initiative aims to replace fragmented Enterprise Resource Planning and other legacy information systems with a harmonized global platform. Broadly speaking, the aim is to reduce the number of operational information systems to less than a tenth of those that existed at the start of the globalization process (a reduction that is significant).

In order to assist standardization on the process and systems fronts, Shell have also sought to instigate a step change in the way that key Master and Reference Data (MRD) is managed in relation to their customers, products, suppliers, materials, technical assets and accounts across the Downstream businesses and functions. One key requirement here is that of deploying quality standards and measures to ensure that key reference data is fit for purpose. This means, for instance, that the right product be delivered to the right customer at the right address, with costs and profits correctly classified and reported. Consequently, Data Quality Standards (DQSs) have been defined along such lines (e.g., no obsolete customers, no duplicate records etc.) and a significant program has been instigated to ensure that streamlined IT systems are cleansed and validated. Cleansing is the process of removing or correcting data that is incomplete, inaccurate or improperly formatted. Validation is the process of ensuring that DQSs have been properly implemented. Again, this is a significant program, with an effort estimated at 300 man-years. Data cleansing is seen as important as poor data quality not only results in inefficient business processes, it also potentially limits organizational ability to analyze, understand and manage the business in the most effective ways.

The effort here mirrors observations in the literature that data quality issues have become increasingly prevalent in practice - costing organizations significantly, alienating customers and suppliers and hindering decision making and the implementation of strategy for example [18, 19, 20, 21, 22, 23]. In addition, data quality in the context of compliance has become more critical since the Sarbanes-Oxley Act of 2002. The intrinsic treatment of data quality (devoid of context) is problematic however [22]. To this extent, a smaller element of the body of literature starts to form the basis of a ‘business case’ for a slightly different but less explored perspective on the issues of data quality. Redman [20] is an early work that notes the difference between comparing data and the real world and ‘database bashing’, which is more in line with typical industrial treatment. Importantly, he also notes that the fact that solution approaches of the former type are often attempted downstream, resulting in improvements are not typically sustained (*ibid*). Orr [24] makes a similar argument, proposing that data quality is “the measure of the agreement between the data views presented by an information system and that same data in the real world” (p.67). Other works take a similarly representational view on data quality, noting the importance of the semantic/ontological foundations of data quality and including incomplete and/or ambiguous representation as key design deficiencies [23, 25]. For reference, the later literature in the area favours the term ‘information quality’ over ‘data quality’. While much of the literature explicitly uses the terms interchangeably, the distinction is that ‘data’ typically refers to the stored content, whereas ‘information’ refers to the situation where such content has been delivered/presented and interpreted [25].

The representational view has relevance here as the MRD team realized that, in the context of streamlining and standardization, data cleansing and validation is not action enough in relation to MRD. The Downstream Data Model (DDM) was thus developed in response to the recognition that the large number of relatively independent projects that were bringing about the transformation in business and IT processes required a standardized basis for integration and consistency across the business. In essence, common processes and common systems indicated a strong requirement for a common data model. The stated business purposes of the model are to:

- Identify the key objects of interest to the business and the relationships between them
- Provide a specification of the information requirements for the Downstream business

- Identify the underlying transactions and relationships
- Provide a basis for checking that the process model includes the processes for managing both objects and data about objects
- Provide a basis for checking that the physical data model, user and system interfaces in applications support the information requirements

In the context of the literature, the approach is a sensible one. The objective of streamlining IT systems around a core set of Enterprise Resource Planning systems is a common means of attempting to provide a seamless integration across a full range of organisational processes – uniting functional and global areas within the business and making their data visible in a real-time manner. Some analyses of ERP systems implementation indicate that organizations must be willing to develop common definitions and understanding for both data and process across the business [26], though typically the concentration is on the link with process.

6.1.2 Foundations of the Data Model

The DDM thus represents a model of the Shell domain that is independent of any system in which representations of the domain may be implemented. This characterization requires a focus on the information requirements of the organization (and thus of any system) allowing the structure or processing of the system to remain undetermined. For reference, the DDM is a Computationally Independent Model (CIM) from a Model Driven Architecture (MDA) perspective. CIMs are relatively under-explored in relation to the work available in other areas of the MDA and initial work in relation to the DDM indicated that a finer separation of concerns was required in relation to the CIM classification. Essentially, one can distinguish an ontological representation (a model of 'what is', in essence a view from nowhere) from epistemological representation (a model of 'what is known' about the domain by some agent, and how it may be represented in a system). In these terms, the DDM is developed as a hybrid model - it is in large part an ontology, but with an epistemological 'gloss', which represents what Shell as an organization (the 'agent') knows, rather than what any particular system knows. Given that business users in Shell are unfamiliar with terms such as ontology and epistemology, the DDM is referred to within Shell as a (conceptual) 'data model'. This can be further distinguished from an implementational representation – a representation in terms of the technology used to implement a particular application or set of applications of the implementation. In traditional data processing this would be known as a physical model.

Given that streamlining within Shell is process-centric, the breadth of the scope was defined as covering the following business processes in Shell's Downstream business (some processes such as Human Resources were scoped out for this version of the model):

- Sell to Business Customer
- Sell to Retail Customer
- Manufacturing
- Manage Lubricants Supply Chain
- Manage Bulk Hydrocarbons Supply Chain
- Procure Goods and Services

The depth of scope of the model was to range from the metaphysical choices at the framework level to a level of abstraction that reflected business language (i.e., leaf subtypes should represent things directly recognized by the business, rather than high-level abstractions of those things). In providing some 'flesh' to the scope, the development of the DDM drew on a range of existing written material as a start point (which meant that interviewing business staff for requirements was not been necessary except for clarification in some cases). The evidence that has been drawn upon in developing the DDM included (a) ISO 15926 [5], (b) the Downstream Process Model, (c) a Glossary of Terms for the Downstream business, (d) the previous version of the Downstream Data Model, (e) Project Logical Data Models (where they have been developed), (f) Physical Data Models from implemented systems, and (g) data from existing systems.

6.1.3 Data Model Development

The work was divided up among several data modellers in the form of schemas. EXPRESS and Visual EXPRESS support the development of a number of schemas that make reference to each other to provide integration. Visual EXPRESS, however, is a single user tool, so that each schema can only be worked on by one person at a time (the tool was thus a determining factor in process terms). Initially, each data modeller was allocated one or more process areas to model as one or more process schemas.

However, it quickly became clear that this was unworkable, because so many things like products, organizations, properties, and locations appeared in many of the process areas without them clearly belonging to one of them. This lead to duplication of concepts between schemas and the need for reconciliation between them.

This commonality of concepts between process areas led to this approach being abandoned in favor of one where:

- Subject Area schemas were developed for common concepts, responsibility for which was given to one data modeller,
- Data modellers were given responsibility for ensuring that requirements from their Process Area were met in the Subject Areas.

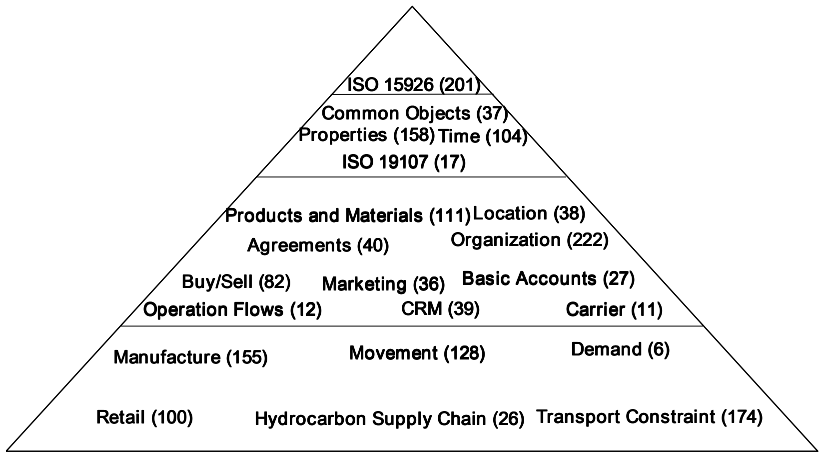

Fig. 30. The set of subject areas for the DDM V2.0

By project completion, almost the whole model was in Subject Areas – this proving an important factor in integrating requirements across the different Process Areas. The final set of Subject Areas is shown in **Fig. 30** – the more abstract and widely referred to Subject Areas are shown at the top, with the more Process Area specific schemas shown lower in the triangle. For reference, the numbers in brackets show the number of entity types in each schema (the total size of the DDM is in excess of 1700 entity types).

7 Conclusions

In this chapter we have brought together several threads of ontological analysis founded on a 4-dimensional ontology and we have presented a case study of their use in a business context. This has demonstrated the effectiveness of an ontological approach to information systems design, in particular for data models.

8 References

1 West, Matthew; Fowler, Julian Developing High Quality Data Models (Version 2.0) 1996 EPISTLE

2 Sider, Theodore Four Dimensionalism - An Ontology of Persistence and Time 2001 Oxford University Press, ISBN 0-19-926352-3

3 Hawley, Katherine How things persist 2001 Oxford: Clarendon Press

4 Stell, John G.; West, Matthew A 4-Dimensionalist Mereotopology pp 261-272 Formal Ontology in Information Systems, Varzi, A.C. Vieu, L. (eds) 2004 IOS Press,

5 ISO 15926-2:2003, Industrial automation systems and integration — Integration of life-cycle data for process plants including oil and gas production facilities — Part 2: Data Model
6 West, Matthew Information Modelling: An analysis of the uses and meanings of associations 2002 PDT Europe
7 Lewis, David On the Plurality of Worlds 1986. Oxford & New York: Basil Blackwell ISBN 0-631-13994-X
8 Suppes, P. Axiomatic set theory 1972 Dover Publications Inc ISBN 0-486-61630-4
9 Aczel, P. Non-well-founded sets 1998 CSLI Publications
10 Gruber, T. R. (1993). A translation approach to portable ontology specification. Knowledge Acquisition, 5(2), 199-220.
11 Ellis, B. Basic concepts of measurement Cambridge University Press 1966.
12 Searle, J.R. The construction of social reality 1995 Penguin Books ISBN-13: 978-0-14-023590-6
13 West, Matthew; Partridge, Chris; Lycett, Mark Enterprise Data Modelling: Developing an Ontology-Based Framework for the Shell Downstream Business 2006 FOMI
14 Moody, D. L. Theoretical and Practical Issues in Evaluating the Quality of Conceptual Models: Current State and Future Directions 2005 Data & Knowledge Engineering, 55, 243-276.
15 Nelson, J., Poels, G., Genero, M. and Piattini, M. Quality in Conceptual Modeling: Five Examples of the State-of-the-Art (Guest Editorial 2005 Data & Knowledge Engineering, 55, 237-242.
16 March, S., Hevner, A. and Ram, S. Research Commentary: An Agenda for Information Technology Research in Heterogeneous and Distributed Environments 2000 Information Systems Research, 11, 327-341.
17 Park, J. and Ram, S. Information Systems Interoperability: What Lies Beneath? 2004 ACM Transactions on Information Systems, 22, 595-632.
18 hengalur-Smith, I. N., Ballou, D. P. and Pazer, H. L. The Impact of Data Quality Information on Decision Making: An Exploratory Analysis 1999 IEEE Transactions on Knowledge and Data Engineering, 11, 853-864.
19 Fisher, C. W. and Kingma, B. R. Criticality of Data Quality as Exemplified in Two Disasters 2001 Information & Management 39, 109-116.
20 Redman, T. C. Improve Data Quality for Competitive Advantage 1995 Sloan Management Review, 36, 99-107.
21 Redman, T. C. The Impact of Poor Data Quality on the Typical Enterprise 1998 Communications of the ACM, 41, 79-82.
22 Strong, D. M., Lee, Y. W. and Wang, R. Y. Data Quality in Context 1997 Communications of the ACM, 40, 103-110.
23 Wand, Y. and Wang, R. Y. Anchoring Data Quality Dimensions in Ontological Foundations 1996 Communications of the ACM 39, 86-95.
24 Orr, K. Data Quality and Systems Theory 1998 Communications of the ACM, 41, 66-71.
25 Price, R. and Shanks, G. A Semiotic Information Quality Framework: Development and Comparative Analysis 2005 Journal of Information Technology, 20, 88-102.
26 Strong, D. M. and Volkoff, O. A Roadmap for Enterprise Systems Implementation 2004 IEEE Computer, 37, 22-29.
27 Chen, P.P. The Entity-Relationship Model – Towards a unified view of data, ACM Transactions on Database Systems, Vol 1, No 1, March 1976, pages 9-36
28 Hay, David. C. Data Model Patterns: Conventions of Thought New York: Dorset House, 1996
29 Partridge, C. Business objects: re-engineering for re-use, Butterworth-Heinemann, (second edition) 2005, The Boro Centre
30 Simsion, G.C. Witt, G.C. Data Modeling Essentials (third edition), Morgan Kaufmann 2005

31 West, Matthew Roles: A Four-Dimensional Analysis pp45-55 Formal Ontology Meets Industry, Eds. Borgo and Lesmo, IOS Press 2008