Ontology meets Big Data: Immutability

Chris PARTRIDGE [a,b,1]
[a] *BORO Solutions, partridgec@borogroup.net*
[b] *Brunel University, Chris.Partridge@brunel.ac.uk*

Abstract. From the perspective of enterprise computing, ontology is seen as a kind detached pure science. When enterprise computing ventures into ontological topics it does not look to ontology to provide it with theories - it devises its own theory-lite solutions. This keynote aims to make a case for joining up these two by identifying an area where enterprise computing can usefully apply ontological theory. It does this using an example; immutability, a current concern in big data. It argues that ontology's theories about change, in particular McTaggart's [1] analysis of ways of viewing time in terms of series, provide a strong explanatory framework for enterprise computing's immutability and have the potential to lead to better solutions. This approach also reveals that there is an aspect of change in computing systems – the epistemic aspect – where a mutable approach (McTaggart's Series A) provides a better explanatory framework.

Keywords. Ontology, Big Data, Immutability, Epistemology, Two-Dimensional Semantics, Bitemporal Data.

Introduction

There is a tendency to assume that pure science (and even more so philosophy) is often so detached from the world that it has no real applications. This is exemplified by Henry Smith's toast to the Mathematical Society of England (reported in Science, 10 Dec 1886); "pure mathematics, may it never be of any use to anyone." Where science and engineering do connect, a typical division of labor gives science the task of generating theories for engineering to apply.

From the perspective of enterprise computing, ontology is seen as falling into the detached pure science category. When enterprise computing ventures into ontological topics it does not look to ontology to provide it with theories - it devises its own theory-lite solutions.

This keynote aims to make a case for joining up these two by identifying an area where enterprise computing can usefully apply ontological theory. It does this using an example; immutability, a current concern in big data. It argues that ontology's theories about change, in particular McTaggart's [1] analysis of ways of viewing time in terms of series, provide a strong explanatory framework for enterprise computing's immutability and have the potential to lead to better solutions. It is a case where theory and utility pull in the same direction providing support for Kurt Lewin's comment [2] (p.169) "There is nothing so practical as a good theory" and so also for the inverse: not having a good theory may be much less practical.

---

[1] Corresponding Author.

This approach also reveals that there is an aspect of change in computing systems – the epistemic aspect – where a mutable approach (McTaggart's Series A) provides a better explanatory framework.

1. Immutability

Immutability is a programming term, and Wikipedia defines an immutable object as "an object whose state cannot be modified after it is created" (http://en.wikipedia.org/wiki/Immutable_object). For example, consider Figure 1. On the left, the Person entity has an 'Age' attribute. Over time this will need to be changed, hence it is mutable. On the right, the Person entity has a 'Date of Birth' attribute. This does not need to be changed, hence it is immutable.

Immutability is recognized to have benefits that simplify programming. One can freely share and cache references to immutable objects without having to copy or clone them. One can cache their fields or the results of their methods without worrying about the values becoming stale or inconsistent with the rest of the object's state. And they are inherently thread-safe, so you don't have to synchronize access to them across threads.

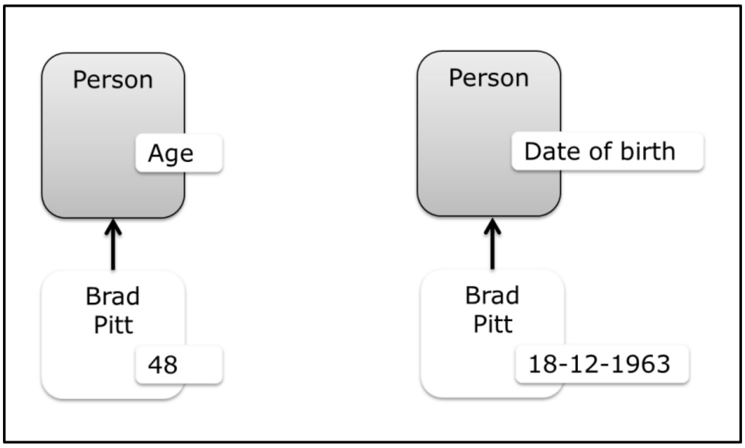

Figure 1. - Examples of mutability and immutability

From the perspective of big data, immutability makes replication simpler. If the data is immutable, then one can only create and read the entries on the database. Keeping replicas up to data can focus on appending data; called 'append-only' – a kind of 'storage monotonicity' with no implication of entailment monotonicity.

Immutability and append-only is not a new notion. It is standard in accounting, where once an entry is made in the accounting records, it is never erased. One can add another entry to correct a mistake, but cannot erase the original entry. If an accountant uses an eraser, he or she is likely to end up in court.

Immutability is normally defined in terms of the way a programmer constructs objects. Given the benefits of this to big data, people are starting to rationalize this in business domain terms. One example is Datomics (www.datomic.com). They propose a way of looking at the domain that makes its objects immutable – and then these can be reflected in the data as immutable data objects. They suggest that changes in the domain can be seen as being in one state at one time and a different state at another time. And they describe how they have implemented a transaction-stamp and append-only structure which reflects this insight in their database; the transaction-stamp gives a monotonic ordering by transaction. However, no substantial theory of change is suggested to back this approach; it is theory-lite.

While the basic idea may be new to enterprise computing, is not new to ontology. It is similar in many ways to what Bertrand Russell [3] an "at-at" approach; at this time it is x, at that time it is y.

2. Ontology's view of change

Ontology has developed a view of change, indeed it is an essential part of any ontological architecture.

2.1. Ontological Architecture

Building a large scale system without some kind of organization guarantees a disaster. Similarly building an ontology for a large scale system without some kind of organization guarantees a disaster. A key aspect of ontology organization is ontology architecture. An ontological architecture needs some organizing principles; it is sensible to make these explicit and consciously and rationally decide on these principles.

Partridge [4] suggests a number of principles organized into components often based upon 'metaphysical choices' –[5,6] subsequently make the same suggestion. Among these are choices that relate to the ontology for change over time.

2.2. Change Architecture Component

Ontology's interest in change has a long history. Around 500 B. C. Heraclitus put forward one view: "Everything flows and nothing abides; everything gives way and nothing stays fixed. You cannot step twice into the same river, for other waters and yet others, go flowing on. Time is a child, moving counters in a game; the royal power is a child's" [7]. His view was that transience is basic and the present is primary.

A generation or so later Parmenides put forward the opposing view: "There remains, then, but one word by which to express the (true) road: Is. And on this road there are many signs that What Is has no beginning and never will be destroyed: it is whole, still, and without end. It neither was nor will be, it simply is—now, altogether, one, continuous…" [7]. His view was that permanence is basic and time is at best secondary, at worst illusory.

Since then there has been a significant amount of research and this is still an active research area.

2.3. McTaggart's Two Change Series

In recent times, McTaggart [1] distinguished two ways of ordering events in time. The first uses a relation of 'earlier than'; this he called the B-series. The second added to the B-series a privileged moment, the present moment; this he called the A-series. These have different formal structures. B-series has no privileged moment, all moments are equal whereas the A-series has a privileged moment – ‘now’, the present – and moments can be classified as ‘past’ or ’future’ in relation to it.

These lead to different relations and properties. A-properties include: present, past, future, 5 years hence, 5 years ago. The B-relations include: earlier than, later than, simultaneous with, 5 years later than. We can see these in **Figure 1.** - Examples of mutability and immutabilityFigure 1; the Person ‘Age’ attribute is an A-property and the Person ‘date-of-birth’ attribute a B-moment.

This gives us two opposing ways to think about the nature of change. An A-series based ‘reality of time’ position where an event changes by first being future, then present, and then past; in other words, by changing positions relative to ‘now’ in an A-series ordering. And a B-Series based ‘unreality of time’ position where events do not change.

The argument that McTaggart puts forward is of the form:

- time is real only if real change occurs
- if there are no changes, then there is no time – time is unreal
- if there are changes, then there is time – time is real

From a pragmatic perspective, McTaggart’s ‘unreality of time’ position has the advantage of providing an explanation for things being immutable.

2.4. Explaining the ‘unreality of time’

However, the position is revisionary – it is intuitively odd. Hence it needs more explanation. One needs to have a story that connects this to our everyday talk about change. There are two complementary ‘stories’, outlined below, that do this.

The first ‘story’ involves redefining talk and can be called ‘change fictionalism’. Fictionalism (in philosophy) is where talk about a thing is useful, but does not commit one to its existence. Metaphorical talk is a common example. This explains why we can tie ourselves in unnecessary knots by insisting that ordinary ways of talking about things (such as changes) are always to be taken literally. Wittgenstein gave this example; if someone says they “married money,” we do ourselves no favors by hunting around for the money they married.

The second ‘story’ involves redefining what ordinary talk about change means. Geach [8] (p.71-72) proposed this and named it “Cambridge” change in homage to McTaggart and Russell who were from Cambridge. He proposed that everyday talk about change had this logical form.

> *an entity x has changed where there is some predicate F that is true of x at a time t_1 but not true of x at some other time t_2.*

The similarities with Russell’s at-at theory of motion mentioned earlier are clear.

The first story explains that we should not take ordinary everyday talk at face value and the second explains how we should interpret it.

2.5. ‘Immutable’ meets ‘unreality of time’

As noted earlier, there is clearly a close match between the aspirations of enterprise computing’s ‘immutable’, and the nature of the ‘unreality of time’. If a system’s ontological architecture’s change component is based upon the ‘unreality of time’, then implementing immutability is natural and straight-forward. The objects in an ‘unreality of time’ ontology do not change so there is no need to represent change in the domain.

The natural next step for the ‘unreality of time’ ontology is towards a four-dimensional ontology where (as Minkowski said in his address to the 80th Assembly of German Natural Scientists and Physicians (September 21, 1908)):

> *“… space by itself, and time by itself, are doomed to fade away into mere shadows, and only a kind of union of the two will preserve an independent reality.”*

This approach to ontology is described in a number of books including [9,10]. Its use in enterprise computing is described in [11].

This four-dimensional approach simplifies the ontology eliminating the need for a direct time-stamping approach as adopted by both Datomics and Geach.

3. Not the whole story of change – two dimensional semantics

McTaggart and four-dimensionalism deal with the changes occurring in the domain that the information system represents. However, this is not the only kind of change that enterprise computing systems have to deal with.

Consider what happens if a user makes a mistake that needs to be changed. They may input ‘Partridge’ as ‘Patridge’ (this often happens to me). The original input needs to be retracted and the correct input made. Consider also what happens when a system makes a best estimate on partial information that gets updated in the light of new information. For example, given what it knows now, my accounting system says my bank balance today is £150. I then tell it about a purchase for £50 on Amazon I made yesterday. It needs to update my bank balance to £100.

There are two types of issue, one where there is mistaken knowledge and the other where there is incomplete knowledge. Both are issues about what the system knows – in other words, epistemic issues. Neither of them are issues about what happens in the domain, they both concern the system’s knowledge of the domain.

3.1. Two-dimensional semantics

This shows that there are two aspects to change; ontic and epistemic. This is a well-known ‘feature’ in both philosophy and enterprise computing. In philosophy it arises in various guises, most recently in one called ‘two-dimensional semantics’ for which there is an extensive literature going back quite a few decades, which includes [12-20].

Kaplan [14,16] has noted that indexicals are good example of how the epistemic dimension works. To see this consider the two statements;

> “I am hungry”
>
> “Chris is hungry on 24th July"

If the first is uttered by me – and so indexed to me – at the right time (on the 24th July) then, they both refer to the same situation. The second requires no such indexing. Typically, indexical statements need to be indexed to a person and a time. It can also need to be indexed to a possible world, if you have adopted a possible world semantics. These are sometimes known as ‘centred worlds’ and coded using a triple of a world, an individual and a time, or a time-slice of an individual in a world.

One dimension of the semantics is ontic; the domain that is being represented. The other dimension is epistemic and is the individual that is doing the representing at a point in time.

3.2. Enterprise computing's two-dimensional semantics

Enterprise computing has a version of two dimensional semantics; bitemporal data [21]. This is data that is associated with a valid time and a transaction time. The valid time denotes the time period during which a fact is true with respect to the real world. The transaction time is the time period during which a fact is entered into the database. Bitemporal data is being incorporated into the SQL standard.

Bitemporal data is called that because it combines both valid and transaction time. The same data will often have different times. Consider a temporal database storing data about the 18th century. The valid time of these facts is somewhere between 1701 and 1800, whereas the transaction time is when we insert the facts into the database, for example, January 21, 1998.

There is obviously a reasonably direct translation between philosophy's two dimensional semantics and enterprise computing's bitemporal data. Valid time is the ontic dimension and transaction time is the epistemic dimension. This illustrates both how the same issues can arise in philosophy and enterprise computing and how they end up with similarly structured solutions based upon the same separation of concerns. It also shows how enterprise computing devises its own solutions within its silo to well-known and researched issues outside the silo. And, if you compare the two approaches in more detail, you will see that bitemporal data is a theory-lite solution; unsurprising given the lack of a suitable theoretical framework in enterprise computing.

3.3. McTaggart's series revisited

The two dimensional semantics ontic dimension naturally relates to McTaggart's Series B. There is a less obvious connection between the epistemic dimension and Series A. It is clear that there is an important sense in which the system's knowledge does change over time. What it knew yesterday could be different from what it knows today, and this could be different from what it will know tomorrow. It seems like 'reality of time' is the natural way to go for the epistemic dimension.

Earlier we discussed 'centred worlds' as a way to think about the epistemic dimension. In a similar way, we can think of 'epistemic reality' as a relation between a momentary state of the individual (system) and what it knows when in that state. The 'what' is the ontology, the view of the domain, and the epistemology is the link to the individual (system) that knows at that moment.

The reality-unreality distinction between the two series captures the different nature of the two dimensions. It is ironic that the unreality of time series (B) should be associated with ontic time and the reality of time series (A) associated with epistemic time, when our natural intuition is to regard ontic time as 'more' real than epistemic time.

3.4. Implementing the epistemic architecture

From an enterprise computing perspective 'immutability' has desirable properties; it is a 'good thing'. However, any implementation has to accommodate both ontic 'immutability' and epistemic mutability.

From a theoretical perspective, the solution would appear to be to treat the epistemology as a view over an ontology, so it inherits its immutability. This treats the two as layers rather than dimensions, which separate what is known from the knowing.

From an engineering perspective, one can order transactions (and time-stamp as well, if desired) and append these increases in knowledge. One can then treat simple retractions as separate transactions. More complex retractions might require more complex processing.

4. Summary

There is an opportunity to join up pure ontology and applied enterprise computing. Both disciplines recognize there is a challenge in dealing with change. Both have seen, though for different reasons, a solution in immutability. Enterprise computing has recognized the practical benefits of immutability, but has no real theory to guide its modeling. Philosophical ontology has a number of theories of change. One of these is based upon immutability (unreality of time). This looks like a close match, which if acted upon could lead, in enterprise computing, to a better understanding of the nature of change, better models and so systems.

However, as we have seen, the situation is more complex. There are epistemic as well as ontological factors to be considered. Here again there is theory from outside enterprise computing that can suggest a solution. Two-dimensional (or two-layered) semantics gives us a better understanding of the issues and a framework from within which we can devise solutions.

References

[1] McTaggart J., The Unreality of Time. 1908. Mind 17.68: 457–474.
[2] Kurt Lewin - Field Theory in Social Science: Selected Theoretical Papers - 1951 - Harper & Brothers
[3] Bertrand Russell - The Principles of Mathematics – 1903
[4] Partridge, C. (2002). LADSEB-CNR - Technical report 06/02 - Note: A Couple of Meta-Ontological Choices for Ontological Architectures. Padova, LADSEB CNR, Italy
[5] Borgo, S., A. Gangemi, N. Guarino, C. Masolo, and A. Oltramari. (2002). "WonderWeb Deliverable D15 Ontology RoadMap." The WonderWeb Library of Foundational Ontologies and the DOLCE ontology. p. 11 - 4 ROADMAP OF MAJOR ONTOLOGICAL CHOICES
[6] Salim K. Semy, Mary K. Pulvermacher, Leo J. Obrst. (2004). Toward the Use of an Upper Ontology for U.S. Government and U.S. Military Domains: An Evaluation. DOCUMENT NUMBER MTR 04B0000063, MITRE TECHNICAL REPORT
[7] Wheelwright, Philip. 1960. The Presocratics. Indianapolis.
[8] Geach, P.T. (1969). God and the Soul,
[9] Mark Heller - The Ontology of Physical Objects: Four-Dimensional Hunks of Matter (Cambridge Studies in Philosophy) – 1990
[10] Theodore Sider - Four-Dimensionalism: An Ontology of Persistence and Time – 2001
[11] Partridge, C. (1996). Business Objects: Re - Engineering for re - use. Oxford, Butterworth Heinemann.
[12] 1979 - Evans, G., Reference and contingency. The Monist 62:161-89.
[13] 1978 - Stalnaker, R., Assertion. In (P. Cole, ed.) Syntax and Semantics: Pragmatics, Vol. 9. New York: Academic Press.

[14] 1978 - Kaplan, D., Dthat. In (P. Cole, ed.) Syntax and Semantics. New York: Academic Press.

[15] 1980 - Davies, M. & Humberstone, I.L., Two notions of necessity. Philosophical Studies 58:1-30.

[16] 1989 - Kaplan, D., Demonstratives. In (J. Almog, J. Perry, and H. Wettstein, eds.) Themes from Kaplan. Oxford: Oxford University Press.

[17] 1996 - Chalmers, D.J., The Conscious Mind: In Search of a Fundamental Theory. Oxford: Oxford University Press.

[18] 1998 - Jackson F., From Metaphysics to Ethics: A Defense of Conceptual Analysis. Oxford: Oxford University Press.

[19] 2002 - Chalmers, D.J., On sense and intension. [consc.net/papers/intension.html]

[20] 2004 - Chalmers, D.J., Epistemic two-dimensional semantics, Philosophical Studies 118:153-226.

[21] Snodgrass, Richard T. (1999). Developing Time-Oriented Database Applications in SQL PDF (4.77 MiB) (Morgan Kaufmann Series in Data Management Systems); Morgan Kaufmann; 504 pages; ISBN 1-55860-436-7

[1] McTaggart JME, Broad CD. 1988. The nature of existence Volume 1. Cambridge University Press: Cambridge ; New York; v. <2 >.

[2] Lewin K. 1951. Field theory in social science: selected theoretical papers. Harper.

[3] Russell B. 1903. The Principles of Mathematics.

[4] Partridge C. 2002. LADSEB-CNR - Technical report 06/02 - Note: A Couple of Meta-Ontological Choices for Ontological Architectures.

[5] Borgo S, Gangemi A, Guarino N, Masolo C, Oltramari A. 2002. WonderWeb Deliverable D15 Ontology RoadMap. The WonderWeb Library of Foundational Ontologies and the DOLCE ontology.

[6] Semy SK, Pulvermacher MK, Obrst LJ. 2004. Toward the use of an upper ontology for US government and US military domains: An evaluation. DOCUMENT NUMBER MTR 04B0000063: 43.

[7] Wheelwright PE. 1960. The presocratics: Indianapolis.

[8] Geach PT. 1969. God and the Soul.

[9] Heller M. 1990. The Ontology of Physical Objects. Four Dimensional Hunks of Matter. Cambridge University Press.

[10] Sider T. 2001. Four-dimensionalism : an ontology of persistence and time. Clarendon Press ; Oxford University Press: Oxford New York; 280.

[11] Partridge C. 1996. Business Objects: Re - Engineering for re - use. Butterworth Heinemann: Oxford.

[12] Evans G. 1979. Reference and contingency. The Monist 62: 161-189.

[13] Stalnaker RC. 1978. Assertion. Syntax and Semantics: Pragmatics 9: 315-332.

[14] Kaplan D. 1978. Dthat. Syntax and semantics 9: 221-243.

[15] Davies M, Humberstone L. 1980. Two notions of necessity. Philosophical Studies 38: 1-30.

[16] Kaplan D. 1989. Demonstratives. In Themes from Kaplan, Almog J, Perry J, Wettstein H (eds). Oxford University Press: Oxford.

[17] Chalmers DJ. 1996. The conscious mind: In search of a fundamental theory. Oxford University Press, USA.

[18] Jackson F. 1998. From metaphysics to ethics: A defence of conceptual analysis.

[19] Chalmers DJ. 2002. On sense and intension. Nous 36: 135-182.

[20] Chalmers DJ. 2004. Epistemic two-dimensional semantics. Philosophical Studies 118: 153-226.

[21] Snodgrass RT. 2000. Developing time-oriented database applications in SQL. Morgan Kaufmann; 504.