What is 'is'?

Marc **de Graauw** <marc@marcdegraauw.com>

Abstract

With the abundance of XML vocabularies a common question is: which seemingly different elements are really the same, and which ones are really different. In business we encounter this problem in data exchange: how do I map the messages and elements from my favorite B2B-vocabulary onto the B2B-vocabulary my trading partner uses? Ontologies try to define which things we speak (or exchange data) about and how we reference them.

In mapping between two ontologies we often use equivalence relationships: 'LastName = given_name', 'Thomas Mann = der Zauberer' et cetera. In the first part of the paper I want to explore some philosophical notions on equivalence: the difference between intension and extension (Frege) and the idea that meanings aren't always precise (Wittgenstein). I will also discuss the relevance for IT of these notions.

In the second part I want to explore some current solutions in XML and Knowledge Management:

1. The naive approach: Let's make a new vocabulary which covers everything, then let everybody use that vocabulary.

2. Adding meta-information: This approach is used in the Context Drivers of ebXML.

3. Published Subject Identifiers (PSI): Make public libraries of unique ID's for things and map to those ID's.

In the third part I will identify some problems in the current solutions and propose an enhancement: we need to capture the knowledge in mappings and we need tools to help reusing this knowledge. An open and standardized format for storing and exchanging knowledge about mappings would be a major step towards ontology interoperability.

1. A bit of Philosophy

Gottlob Frege (1848 - 1925) is generally seen as the founding father of modern logic and the greatest logician since Aristotle. Frege investigated 'meaning' and 'reference' as founding blocks of logic. Of special interest is his study of equivalence statements. Frege asks what the nature is of a statement such as:

'The morning star is the evening star.'

When this statement reflects an identity between two objects, it would seem to be equivalent to:

'The morning star is the morning star.'

Since the morning star is the evening star, we may replace the phrase 'the evening star' with the phrase 'the morning star'. The statement would simply say that an object is identical to itself. This can't be the nature of the equivalence relation, according to Frege. The first statement conveys knowledge (it states something we did not know before), the second is a mere tautology.

Frege therefore proposes a distinction between the intension (Sinn in Frege's work) and the extension (Bedeutung) of a term. The extension of a term is what it refers to - in the case of 'the morning star' the extension is the planet Venus, as it is in 'the evening star'. The intension is an inner concept, an idea we have which corresponds to the term. Intension is also called connotation, extension is also known as denotation. For Frege, the intension is something shared between speakers (different speakers have the same intension) though he recognizes that in natural language the intension might be different for different speakers. The intension of 'the morning star' and 'the evening star' is different - the former has to do with a bright light we sometimes see in the morning sky, the latter with a bright light in the evening sky. The nature of the equivalence statement 'The morning star is the evening star' then is a relation between intensions. The statement says those two different intensions have the same extension (the planet Venus). In this way the equivalence statement can convey useful knowledge. Since the intensions of 'the morning star' and 'the evening star' differ, the equivalence statement adds a fact to our body of knowledge.

For Frege, a word has an intension, and to an intension belongs an extension:

word -> intension -> extension

An example could be:

'Venus' (the word) -> The second planet of the Sun -> Venus (the planet)

Frege's scheme has been widely followed since. Frege uses it mainly for proper names or similar expressions which refer to a single thing. It has been widely extended to incorporate all kinds of nouns (Frege himself handled most nouns in a different way). Nouns are more complex than proper names, for the extension is no longer an individual thing, but a collection of things.

Wittgenstein developed ideas quite similar to Frege's in his early work, the 'Tractatus logico-philosophicus'. He believed he had solved all philosophical problems in this book, and consequently stopped practicing philosophy. Later he returned to philosophy to become the greatest critic of the younger Wittgenstein. Most pointedly, he denounced the simple assignment of names to things. Language is much richer than this simple scheme. There are many kinds of sentences, and the name-thing connection only applies well to a limited subset of all language expressions. It does not reflect commands, poetry, jokes et cetera very well.

The later Wittgenstein also combats simplicity in meaning. There is no such thing as a fixed meaning for many kinds of words. A word like 'game', according to Wittgenstein, has no fixed meaning. There is no characteristic common to all ball games, board games, card games et cetera. Most games are competitive, but some are not. Most games involve multiple players, others only one. So a word like 'game' has no fixed meaning with a fixed set of characteristics. It is more like a 'family' of characteristics. Games have characteristics out of this 'family', but they don't necessarily have all of those characteristics. Moreover, according to Wittgenstein the list of characteristics is not fixed for all time. The meaning of a word may change in different circumstances and times. The meaning of a word depends on the circumstances it is used in. Wittgenstein says: 'the concept "game" is a concept with hazy edges'[WITT]. This important notion ignited a whole new school of language philosophy, known as 'ordinary language philosophy'.

2. Back to Information Technology

So what is the relevance of these philosophical notions to IT? We have lots of data and descriptions of those data. Take for instance the abundance of vocabularies for B2B exchange like xCBL, FinXML, FpML et cetera. Those vocabularies can be seen as ontologies. Older EDI technologies such as X.12 and EDIFACT are also ontologies. Beside those 'industry-strength' solutions, there are lots of tailor-made data exchanges between companies, often using nothing more than simple ASCII comma-separated files. Together with their documentation, those ASCII-files also constitute ontologies. And even within larger companies many different ontologies exist within the different legacy databases of the different departments. Those different data sources present huge interoperability problems.

First some terminology. In this paper I will frequently talk about vocabularies. By a vocabulary I mean a set of definitions of data items (fields in a record, elements or attributes in XML) and/or associated messages. It is important to distinguish between the data definitions and the data themselves. Vocabularies are about the definition of data items such as 'person' or 'money'. Actual occurrences in a data store, say 'Marc de Graauw' or '$20,00' are not part of the vocabulary. Ontologies are the same as vocabularies but may incorporate more structure, such as class relationships. When the difference is irrelevant, I will use the two interchangeably. Interoperability is the problem of communicating between sources with different ontologies. Mappings are one-on-one translations between two ontologies, which enable translation from a source ontology to a destination ontology.

A simple example of different ontologies constitutes the concept of 'money'. It might seem a relatively clear-cut concept, but you might think of dollars while I think of Dutch guilders (soon to be Euros...). An economist might wonder which definition of 'money' I use: M0, M1, M2, M3, M4... An anthropologist could include shells and beads as money. An historian could think of ducats or Roman sestertii. An accountant will wonder whether my money includes VAT or not. So when we need to exchange data between ontologies, we need to take care that what we call 'money' is actually the same thing in both ontologies.

One of those interoperability problems is finding out which data items from different sources are the same. To do that, we need to compare the meanings of those data items. The simplest way of seeing whether two items are the same is comparing their extensions. However, comparing extensions is often impossible. Consider a statement like 'All primes between 10 and 20.' The extension is easy to establish: {11, 13, 17, 19}. But in most cases a simple enumeration is not possible. How could we enumerate all occurrences of 'money' or 'person'? Even in a more limited context, a concept like 'employee' is difficult to enumerate. It might be possible to do this for a fixed moment in time within a single company, but employees change as time goes on. Definition by establishing the extension directly is therefore in most cases impossible.

So when comparing the extension is not feasible, we will have to compare intensions to achieve interoperability. This means we have to look up data definitions (metadata) for different data sources and compare those data definitions. Comparing human-made definitions is a though job. Different companies of departments may come up very different definitions for things that really are the same, and with very similar definitions for things that are very different in reality. Most important here is Wittgenstein's view that meanings aren't precise and fixed. This is true of natural language but no less so for data definitions in an IT context. First of all, hard as we try, mistakes and obscurities occur in what we write down in our data definitions. Second, in making data definitions we may find that a lot of data aren't that well defined to start with. In other words, when we make data definitions for a data source this sometimes is the first attempt to define the data at all, and when there already is a definition, it is often not precise enough. Third, when we make a definition like 'an employee is a person working at a company', we introduce many new words ('person', 'work', 'company') from natural language. When meanings in natural language aren't precise, those definitions aren't going to be precise either. So the most important lesson is that we must take Wittgenstein's critique serious and accept that meanings aren't precise, not in the 'real world' and not in IT. That doesn't mean we should give up on making meanings precise - we must always try, and making meanings more precise is always a useful job. We should just not think we can fix meanings once and for all in any but the most limited contexts.

3. Some Current Solutions to Interoperability

In this part I want to explore current solutions in XML and Knowledge Management. The first, which I shall call 'the naive approach' is: Let's make a new vocabulary which covers everything, then let everybody use that vocabulary. Probably everybody thought of this sometime and found out it does not work in practice. Multiple vocabularies are a fact of life. In a limited context such an undertaking is feasible, but the amount of work involved shouldn't be underestimated. An amusing and telling story goes back to the early nineties. Current wisdom for large companies at the time was to make a 'corporate data dictionary', containing metadata about all databases in the company to facilitate interoperability within the company. One company did exactly that. A team was set up to make and fill the corporate data dictionary. After a short while, the CEO visited the group to see how things were proceeding. Everything was fine. The group was on schedule, had already stored a vast amount of metadata in the corporate data dictionary, and the first benefits were already being reaped. After a while though, little was heard of the team anymore, and status reports showed little progress. Another visit revealed the problem: the team was spending all of its available time keeping metadata already in the corporate data dictionary up to date, and there was no time left to add new metadata. As company divisions changed their databases to meet new business needs, the metadata had to be updated too. One cannot underestimate the amount of work involved in keeping metadata up to date.

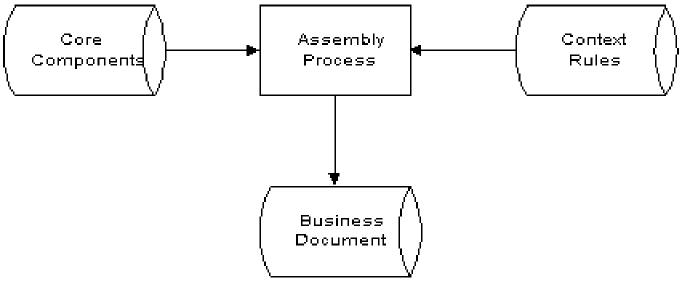
Another approach to interoperability is the use of Published Subject Indicators [1] (PSI) as used in Topic Maps. The basic idea is to make public libraries of unique ID's for things. In our vocabularies we incorporate PSI's, and then we can compare the terms in our vocabularies. In an informal example:

```
Topic: 'United States of America'; PSI: US
```

```
Topic: 'Verenigde Staten van America'; PSI: US
```

The PSI's in the English and Dutch topics allow us to conclude that both topics are the same. Note that this really just shifts the problem from vocabularies to public libraries. In general we can say this approach is successful if the problem space consists of clearly delimited entities and there is a widely accepted canonical public library. Examples of areas were this approach will work are for instance ISO currency and country codes.

Yet another approach is taken in the ebXML initiative. Within ebXML a set of Core Components is defined. These Core Components are primitive data types to be used in B2B data exchange. However, the Core Components do not have semantics, they are only syntactical constructs. The core Components are used to 'assemble' Documents which can be used in B2B exchanges. The assembly process is rule-based. Industries or businesses can define Context Rules which say how Core Components are transformed and aggregated into a Business Document. In a Business Document semantics is added. In a simplified [2] picture:



Context Rules use Context Drivers which add information about the context. ebXML uses the following proposed Context Drivers[CCDRIV]:

- Region (Geopolitical)
- Industry
- Business Process

[1] In the original ISO Topic Map specification, these are called 'Public Subject Descriptors'. The reviewers of my proposal also mentioned 'Formal Public Identifiers' as used in SGML. Since I don't know SGML, I will gladly include it here on their authority.

[2] This is simplified because ebXML uses more items in the assembly process, such as Assembly Rules. Those are not relevant to this discussion.

- Product
- Official Constraints
- Role
- Temporal
- Information Structural Context
- Application Processing
- Service Level
- Business Purpose
- Virtual Marketplace
- Contractual

With the use of these Context Drivers and Context Rules, the Assembly Process can make a different 'address'-component for companies in the USA or the Netherlands with the use of the 'Region' context Driver, and still use Core Components as the building blocks.

The main advantage of this approach is that interoperability is facilitated because all Business Documents use the same Core Components. When two Business Documents need to be mapped, we can look up which items originate from the same core Component, and use the Context Rules to determine how the transformation between the two items has to take place. This process could also be (partially) automated. The main drawback is we will have to rewrite existing ontologies if we want to use this solution.

4. Problems with Current Solutions

In the this section I want to look at problems with the current solutions and propose an enhancement.

In an interesting survey of existing approaches, Wache et. al.[WACHE] distinguish three main types of solutions to the problem:

1. Single Ontology approaches. All data definitions are made in a single unifying ontology.
2. Multiple Ontologies. All data definitions are in their own ontology, and mappings are provided between those local ontologies.
3. Hybrid Approaches. Local ontologies are maintained, but their vocabulary is drawn from a global shared vocabulary.

In this classification, the ebXML approach is a Hybrid Approach. PSI's as used in Topic Maps could also be seen as a Hybrid approach [3]. The drawback of Single Ontologies and (though less so) Hybrid Approaches is that we have to redefine existing ontologies. Those solutions therefore are only applicable when we are in a position to (partially) redefine existing ontologies. This is often not the case in B2B contexts. This does not mean that the work involved in creating new unifying ontologies is useless: on the contrary, when new ontologies are introduced that are widely accepted, this is a major step ahead for interoperability. So efforts towards unification should be continued. It is important though to realize multiple ontologies are, and will remain, a fact of life, and we will always need ways to achieve interoperability between multiple ontologies.

The major plus of Multiple Ontologies is that existing ontologies remain in place, and only mappings between those local ontologies have to be provided. The number of needed mappings however grows exponentially with the number of local ontologies. In part this could be fixed by using intermediate ontologies in defining mappings: we translate from a source ontology one to an intermediate ontology, and then from this intermediate ontology to a destination ontology. When we need to add a new local ontology, all that is needed is a mapping to the intermediate ontology and interoperability is possible with all ontologies which already have a mapping to the intermediate ontology.

There is also research underway towards automated integration. Automated integration too can never be a solution to the entire problem space. There will be a need for human intelligence to create the 'end points' for automated integration. That is, before automated integration can take place, existing ontologies have to be replaced by new ontologies which enable automated integration. And even when existing ontologies have been replaced, we will still need human experts to adapt those ontologies to changing business needs. The ebXML solution with context drivers could make partial automated integration possible. Automated integration is interesting, but manual integration is still state of the art.

5. Capturing the Knowledge in Mappings

Now back to a 'real world' example. In actual mappings between ontologies, we often do not really establish semantic equivalence in a true sense as needed in for instance Topic Maps PSI's. When we have found we can use 'CustomerAddress' of our trading partner as the 'billing_address' in our online billing application, we stop. We do not need to find out whether they are truly equivalent in all circumstances. There is no direct business need to find out whether they are equivalent in all situations, and therefore the boss doesn't pay to find this out. Solutions like PSI's do not work here, because PSI's require true semantic equivalence. The interesting observation here is that most real world mappings are unidirectional: we translate from a source ontology to a destination ontology for a specific business process. For instance, an order goes from buyer to supplier. It does not go back (though a different document such as an invoice or order confirmation might go back). So for an order only a translation from the buyer's ontology

[3] The use PSI's in ontologies can also be seen as Multiple Ontologies where the mapping to a third ontology (the public library of PSI's) is incorporated in the original ontologies.

onto the supplier's ontology is needed. This unidirectional nature of business exchange means that often we do not establish equivalence relationships, but subset relationships between ontologies. In the above example, 'CustomerAddress' is a subset of 'billing_address'. All instances of 'CustomerAddress' constitute a valid instance of 'billing_address'. We do not know whether the reverse is true, and in this example we do not need to know either.

Another problem altogether is the reliability of mappings. When we have to map our ontology onto another, and we have PSI's included both in our ontology and in the other one, how do we know we can trust the PSI mapping in the other ontology? Simple inclusion of PSI's doesn't tell the whole story, we also need to know whether this was done as a quick job for a particular occasion, whether it was done by an expert et cetera. Yet another problem occurs when the other ontology does not contain PSI's, and we do not control the other ontology. In that case we cannot incorporate PSI's in the other ontology. (Topic Maps provide a solution for this through Merge Maps, where we can express which items we want to map.) All this might sound as if I disapprove of PSI's, but I want to stress again that I do not. PSI's (and unifying ontologies in general) are a good thing, and we should pursue clear and widely accepted unifying ontologies whenever possible. The only point made here is PSI's and unifying ontologies do not provide the whole answer.

It might be tempting to conclude that we simply have to make a mapping between every two ontologies we use. That, however, is going to far. Even when we do not always establish true semantic equivalence relationships, the mappings we make are certainly for a great part reusable. What we need to do is capture knowledge about the mapping process itself. We need to store the fact that we can use 'CustomerAddress' as 'billing_address' in this particular context. Then, when someone else needs to find out whether 'CustomerAddress' can be used as 'InvoiceAddress' in a different context, they can use this information. When we store this kind of information, we could facilitate the process of mapping ontologies through the use of semi-automated tools which show existing mappings for items in our ontology that we need to map onto another ontology. The human expert making the mapping can still make all the relevant choices and provide new mappings where existing ones can't be reused. Such semi-automated tools could then generate a new mapping, which also can be stored to provide information for the next one. It would also become much easier to exchange information about mappings without having to provide full one-on-one equivalence relationships.

So we need to capture knowledge about the mapping process itself:

- Who asserts that this is a valid mapping?
- What is mapped (source and destination data items)?
- What is the kind of relation (equivalence/subset/superset)?
- What is the context of this mapping (which Context Drivers of ebXML apply)?

We will need to be able to make mappings between ontologies in different formats (XML, Topic Maps, DAML+OIL, ebXML, RosettaNet, EDIFACT, proprietary and legacy formats) in different storage formats (local files, web resources, databases, paper). Topic Maps would provide an excellent vehicle to store such mapping information, though RDF and especially DAML+OIL could do the job as well.

When we store knowledge about the mapping itself, we must allow inconsistencies to occur. For instance, John might assert that 'CustomerAddress' is equal to 'billing_address', Paul might assert that 'billing_address' is equal to 'InvoiceAddress' and George might assert that 'CustomerAddress' is not equal to 'InvoiceAddress'. All this knowledge is useful in making new mappings, even when it is clear that at least one of our existing mappings is wrong. It makes a difference whether UN/CEFACT asserts that an item from EDIFACT is equal to an item from ebXML or our neighbor next door does (UN/CEFACT made EDIFACT and took part in making ebXML).

Using a model such as this we should be able to store the mappings themselves, store metadata about the mappings (such as who made them), retrieve all existing material about an existing ontology and signal inconsistencies. In particular an open and standardized format for storing and exchanging knowledge about mappings would be a major step towards ontology interoperability.

6. Conclusions

1. We need to realize meanings are imprecise, in ontologies as well as in the real world.

2. Multiple ontologies are a fact of life, so we need mappings between multiple ontologies.

3. Making mappings will be an act of human intelligence for the near (and possibly quite distant) future.

4. Mappings will often be imprecise and partial, made for one specific business goal and not for general purposes.

5. We need to capture the knowledge incorporated in human made mappings for reuse.

6. We need an open and standardized format for storing and exchanging knowledge about mappings.

Bibliography

[FREGE] Frege 'Über Sinn und Bedeutung' ('On Sense and Reference') in Gottlob Frege, 'Funktion, Begriff, Bedeutung', Goettingen 1962. This remakably clear and concise essay is the best introduction to Frege's semantics.

[WITT] Wittgenstein, 'Philosophische Untersuchungen' ('Philosophical Investigations') in Wittgenstein, 'Werkausgabe band 1', Frankfurt am Main 1984. One of the classics of 20th century philosophy, and a 'must read' for everybody interested in philosophy of language.

[WACHE] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann and S. Hübner, 'Ontology-Based Integration of Information - A Survey of Existing Approaches', in

'Proceedings of the IJCAI-01 Workshop: Ontologies and Information Sharing', Seattle, WA, pages 108-117 (http://www.tzi.de/buster/papers/SURVEY.pdf)

[EBCCDOC] 'Document Assembly and Context Rules v1.04', UN/CEFACT and OASIS 2001 (http://www.ebxml.org/specs/ebCCDOC.pdf)

[CCDRIV] 'Catalogue of Context Drivers v1.04', UN/CEFACT and OASIS 2001 (http://www.ebxml.org/specs/ccDRIV.pdf)

[EBCNTXT] 'Context and Re-Usability of Core Components v1.04', UN/CEFACT and OASIS 2001 (http://www.ebxml.org/specs/ebCNTXT.pdf)

[XTM] 'XML Topic Maps (XTM) 1.0', TopicMaps.Org, 2001 (http://www.topicmaps.org/xtm/1.0/)

[TMISO] 'ISO/IEC 13250 Topic Maps', ISO/IEC 1999, (http://www.y12.doe.gov/sgml/sc34/document/0129.pdf)

Glossary

PSI	Published Subject Identifiers

Biography

Marc **de Graauw**
Consultant
Independent Consultant
Amsterdam
Netherlands
Email: marc@marcdegraauw.com

I am Marc de Graauw, born 1961, and I live and work in Amsterdam, the Netherlands. I graduated cum laude in Philosophy from Nijmegen University in 1989. As a student of Philosophy, I studied Philosophy of Language and semantics in particular. My thesis was on the reference of nouns of tangibles, i.e. what does a word like "chair" or "cat" refer to.

After university I started working in IT. In 1998 I continued working in IT as an independent consultant, specializing in intercompany data exchange, first in a traditional EDI environment, but soon in XML and Internet environments. I have published several articles (in Dutch) on XML en e-commerce and have been a speaker on conferences in the Netherlands, and organizer of a workshop on XML and e-commerce.

It struck me that often the problems we face in IT are very similar to the problems I faced as a student of Philosophy. In my intellectual work I want to explore these problems and hope to apply the conceptual clarity and analytical skills I admired in Philosophy to the problems in IT.