

# Semantic Overlays Over Immutable Data to Facilitate Time and Context Specific Interoperability

Pete Rai ([peterai@cisco.com](mailto:peterai@cisco.com)) - Principal Engineer - Cisco

Stephen Tallamy ([stallamy@cisco.com](mailto:stallamy@cisco.com)) - Engineering Architect - Cisco

*The authors have led teams who have developed a Semantic Interoperability engine in the context of video and television. This uses semantic overlays to provide same-as services over multiple, differing, disconnected but semantically related datasets. The engine is used to power [locatetv.com](http://locatetv.com). It makes interoperable connections between people, shows, channels, service providers, sports, media, etc. The techniques used have a wider applicability especially in context of IoT use cases.*

## Overview

When people discuss Semantic Interoperability there is often a tendency to frame it in the context of “making things the same” from the ground up. This manifests itself as the promotion and adoption of standards for data types, measurement units, data formats, exchange documents, file encodings, containers formats, etc.

Whilst this is a laudable goal in general, when applied to wide domain like IoT, it can become a limiting factor. A stipulation of “sameness” in order to interoperate can often be disruptive for on-the-ground use cases and agendas. The volume and complexity of changes required can introduce significant delays and push interoperability to a far horizon. This in turn encourages a work-around or make-do approach for facilitating ad hoc semantic connections.

An alternative approach is to accept the “messy edge” of IoT for what it is and apply semantic interoperability layers over-the-top, as and when they are needed. This approach is specifically designed to leave the source data elements untouched and effectively immutable.

Source data is structured the way that it is for good reasons, usually linked to specific use cases. The fact that interoperability with other systems calls for reinterpretation of some data should be treated

as a time and context specific special case. Indeed each “interoperable moment” may call for a new and different interpretation of the source data.

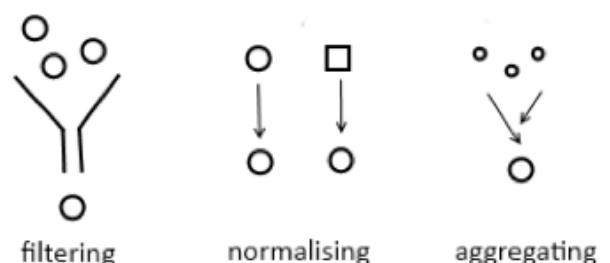
## Domain of Consideration

We can call a specific context and time a “Domain of Consideration” and we should recognise that many such domains can exist simultaneously, each calling for distinct interpretations; different interpretations of data, at different times for different reasons. Hence it’s preferable to leave the source data alone and treat interoperability interpretation as an overlay problem.

Take as an example multiple edits of the same movie, say a full version and also a censored airline edit. For some people, like a person publishing a review or a rating, these can be treated as the same. However for others, like an airline with young passengers, it’s vital to know that these are different. Both players here have a different domain of consideration. Both are equally right. No *single* set of semantic interoperability operations can simultaneously satisfy both their needs.

## Preserving Meaning

When trying to force data into any given interpretation, we inevitable will end up destroying some of its semantic meaning. Some typical operations used to shape data into a form that is interoperable are shown below:



filtering

normalising

aggregating

In one-size-fits-all models, it is typical to settle on a lowest-common-denominator interpretation of the source data.

For example, several movie rating systems exist; some from zero to four stars, others from one to ten, some allow half stars, others only whole stars, some are percentages, etc. (One can see how this problem has equivalence for many classes of IoT sensors, like thermometers, pressure gauges, et al).

It would be naive to expect all players to adopt a single standard for movie ratings. The changes required at IMDB, Rotten Tomatoes, iTunes, etc. would be immense. In many cases, the rating scheme being used is integral to the originating brand and feature set. However, it is easy to see how a collective interpretation across these datasets would be of value. So here we need to apply a normalising function to be able to gain interoperability between them. However, we must accept that this will inevitably destroy some meaning.

It would be a bold claim that one normalising function is correct for all use cases. Time and context will inevitably mean different normalisers are valid for different domains of consideration. Hence, this problem of normalised interpretation lies outside (or over-the-top) of the immutable source data. The same is true for other data operations such as filtering and aggregation.

Further, we can foresee how interpretations of data can change over time. Generally, early interpretations tend to be naive and work-a-day. As systems grow and mature, interpretation of the data can become more specialised and nuanced. This is especially true when taking into account the variety of data that a system will experience over a long period of operation.

This temporal dimension means that normalising functions that were once sufficient, maybe viewed as incorrect over time. If we apply one-time semantic interoperability, it means that it will not be possible to re-interpret this historical data after-the-fact. In effect, some meaning will have been permanently destroyed.

In contrast, over-the-top semantic interoperability readily allows for temporal re-interpretation of source data. It is simply a matter of replaying the normalisation and matching steps with improved rules. This temporal dimensionality can be thought of as just another domain of consideration.

### **Predicate Based Interpretation**

The idea of using predicates to model knowledge is well understood from the realm of computational semantics. Here all knowledge is modelled as a series of triplets of the forms:

[entity, predicate, value]  
[entity, predicate, entity]

Here the *entity* is a domain specific object like an [actor] or a [movie]. The *predicate* is an identifier of a property of a given class of entity, such as [date-of-birth], [gender], [running time], [genre], etc. Finally the right hand side is either a specific *value* or another *entity* within the domain.

Applying predicate based modelling can help in interoperability scenarios by using a given set of predicates to define the semantic overlap that is required by the use cases being considered. This set of predicates can be subset of all the predicates that are known and associated with entities within the domain.

Such a set of predicates form a de facto “interoperability contract” between two or more sources of data; an intent to interoperate over this canvas. What is then further needed is for each actor to supply its pre-predicate normalising functions to honour this interoperability contract.

### **Facts, Opinions and Trust**

It is important to realise that not all predicates can be interpreted the same way when considering the factual truth of their values. That is because the underlying source data may be subject to a various judgements, viewpoints, estimations, etc.

We can divide the predicate space into two separate and distinct classes, *positive predicates* and *normative predicates*.

A positive predicate represents a data element for which there is a singular true value in existence. Examples of positive predicates are [date-of-birth], [awards-won], etc. Normative predicates are used when there is no possible notion of a true value. Examples of normative predicates are [rating], [genre], etc.

Just because positive predicates have a true value, it does not mean that any given positive statement is true. Whether we believe it to be true is based upon our trust in the originator of the statement and how often the same statement has been made by different, independent sources. Normalising functions over positive predicates must aim to preserve the underlying truth of the values being represented.

In contrast a normative statement is never true in any strict logical sense. Normative statements gain value by trust in the originator's judgement and by a consensus amongst a community of users. Normalising functions over normative predicates are more able to apply interpretation over the values being represented.

In terms of semantic interoperability, an overlap in a set of positive predicates can be sufficient to declare the underlying entities to be the "same" for a given domain of consideration (even though they may not be the same in a wider context). In contrast, normative predicates can never be used to determine sameness or equivalence of entities.

For example, a set of positive predicates can be used to define "sameness" of a movie between two different data vendors for a given domain of consideration. Perhaps the [title], [year-of-production] and some [primary-cast] overlap will be sufficient. This semantic interoperability contract can be applied to say that a movie in IMDB is the same-as a movie in Wikipedia. Other players may have use cases which demand a more rigorous contract that includes perhaps [running-time] and [certification].

## Conclusions

Semantic Interoperability can never be just a matter of imposing schemas, standards and lexicons on the source data. This approach is

inflexible and cannot hope to span all use cases. It inevitably leads to implementation delays, destroys meaning and ignores the competing agendas of the actors.

Treating interoperability as an after-the-fact problem allows for greater flexibility. It means that different semantic overlays can be applied in different contexts and different times (or even simultaneously). This methodology leaves the source data unchanged and treats it as immutable. It only mandates that interoperability occur over some defined semantic canvas for a given domain of consideration. This canvas is likely to be a subset of all the knowledge that is present in the wider systems.

Using predicate based modelling provides a ready model to define contracts for interoperability between systems. Here a predicate can act as a common system of interpretation over-the-top of source data. Well formed sets of data operations such as filtering, aggregating and normalising can be used to map data into the interpretations that the predicates demand.

Understanding the nuisances of positive and normative predicates allows for declarations of equivalence between entities within the context of a domain of consideration. It can be used in conjunction with trust model to enhance the belief and veracity of interoperability decisions.

## More Information

The authors can supply more information about the approaches outlined in this short paper. If you would like to know more, then please do contact them both via the email addresses listed above.