

On Property: Property vs. Attribute

-- Ontology representation language: OWL--

Riichiro Mizoguchi, ISIR, Osaka University

Abstract: You must be careful in using the term “property”, since there are multiple meanings in “property” as discussed below. It is a bit dangerous to directly use OWL as an ontology representation language because it is an assembler or an interlingua for ontology interchange.

1. In logic, they call unary predicates a **property**. That is, “*human*” in human(X) is a **property** just like “*red*” in red(X). Although *human* is classified as a Sortal and *red* is as nonsortal to distinguish them in Sortal logic, it is not enough to avoid confusion.
2. In OWL, they call links between two nodes **property**, since OWL is designed by bringing the concepts used in semantic network into its conceptual framework. I could say, OWL conceptualization might lead users to lower level of abstraction at which nodes and links are the main conceptual constructs close to implementation level. In contrast with logic, OWL **property** corresponds to a binary predicate, while it does to a unary predicate.
3. In AI, they clearly distinguish between **property** and **object**. The former is necessarily associated with the latter which in turn cannot exist without any property. That is, *human* is an object and *tall* is a **property**.
4. In ontological engineering, they share a lot with AI people in the use of **property**. Furthermore, they believe an ontology has to be representation-independent, even from logical expression.

Logicians might say X in p(X) stands for an individual before class identification, so there is no difference between *human* and *red* because both are predication of what X is. Ontology researchers who use logic extensively have to accept this position and they follow the terminology in logic with sortal logic. But, Some of the ontology researchers including me use logic as a secondary dealing with ontological reality as the primary and do distinguish the two.

5. The main issue: **property** and **attribute**
There exists serious terminological/ontological confusion between **property** and **attribute**. Ontologically, “being *red* of a rose” and “the *color* of a rose is red” are different. It is apparent when you see red(rose) and color(rose, red). Ontologists do not care about the terminology, however, the differentiation between the two are essential. According to the logic terminology, let us call the former **property** and the latter **attribute**.

Now we are ready to explain how OWL is inappropriate for end users to write/design an ontology. OWL does not help users distinguish among *role*, *relation*, *attribute*, *action*, etc. It only provides nodes for concepts and property(link) to tie them together. It is apparent from that *teacher*, *left-of*, *color*, *buy* are intrinsically different from each other in spite of the fact that they can be represented by binary predicates. Ontologically, there are many different things which can tie two nodes with different meanings. Even worse, *OWL: property* is very misleading, since it reminds us of *unary property* defined in 5. What you can represent by *OWL:property* is **attribute** rather than **property**.

Furthermore, some people represent *roles* by *OWL:property*, which is incorrect, of course. A role is not a binary relation or an attribute. In short, it is not what can be represented by *OWL:property*. Let us take *father* as an example. Father is not a relation. What is a relation is *father-of*. *Father* is a role and represented as father(X) which is intrinsically dependent on children and defined in the context of the relation: father-of(X, Children).

It is worth to pay a close attention to what people mean by **property**!