Should Role Concepts be Included in a Taxonomy? Working Note 11

Peter Clark
Knowledge Systems
Boeing Applied Research and Technology

Jan 27th 1998

1 Introduction

This is a brief working note for internal discussion.

It is common to distinguish concepts which can "exist on their own" (eg. `Person`, `House`) from concepts which are, by definition, in relationship with other concepts, and whose properties depend on that relationship (eg. `Teacher`, `Producer`)[1]. (Sowa refers to these as concepts with "firstness" and "secondness" respectively). These second type of concepts we refer to here as "role concepts". The question we consider here is: should we reify and place role concepts in the global ontology we are building, or should we handle them differently using relations? The problem posed by role concepts is that it is often very difficult to place them taxonomially. For example, consider `Producer` – a `Person` can be (but it not necessarily) a `Producer` (so what is the taxonomic relation, if any?). Similarly, a `Pump` in an aircraft's hydraulic system is a `Producer` of hydraulic pressure, and/or of energy, although we could also view it as a `Consumer` of energy (from the `Engine`), or a `Converter`. Similarly, a `HotDog` can be seen as producing energy (for the body), or producing income for a vendor, although it consumes money for the purchaser etc. The problem here is it is often difficult to intrinsically classify objects as being in a particular role – rather, it depends on how we view the situation, or more formally, which model we choose to use to describe the situation.

2 Example

As an example, consider a simple notion of production, which involves the creation of a product which is then consumed by a consumer. This model occurs in aircraft hydraulics in several ways, for example we can think of a pump as a producer of energy, which is then consumed by an actuator; we can also think of the pump as a producer of vibrations, which are 'consumed' by a dampener. Figures 1 and 2 show two alternative formulations of production, the first where roles have been refied into concepts, and placed in the taxonomy, and the second where roles are represented as relations (between objects and models in which they participate).

[1] In fact, we are already in trouble: Pat Hayes disputes that such a distinction can be made, claiming no concept "exists on its own" (eg. all `Person`s necessarily have a`BirthDate`). However, we will perservere.

```
;;; ------------------------------------------------------------------------
;;;             MODEL OF PRODUCTION
;;; ------------------------------------------------------------------------
;;; "If a producer is producing a product for a consumer,
;;;  Then the producer supplies the product to the consumer, and sends
;;;  the product, and the consumer recieves it."
Forall P:Producer, R:Product, C:Consumer
        produces-for(P,R,C) -> supplies(P,R,C) & sends(P,R) & receives(C,R).

and also taxonomically (where <| means "has superclasses")
        Producer <| Device           ;;; (say) This will do for now.
        Consumer <| Device
        Product <| Entity

;;; ------------------------------------------------------------------------
;;;             USE OF THAT MODEL FOR A HYDRAULIC SYSTEM
;;; ------------------------------------------------------------------------
Instances are Pump1, Actuator1, Dampener1, etc. We place these in the
taxonomy as follows:
        Pump1:Producer            ;; ie. Pump1 is of type (ie. class) Producer
        Actuator1:Consumer
        Dampener1:Consumer
        Power1:Product
        Vibrations1:Product

Then describe the system:
        produces-for(Pump1, Power1, Actuator1)
        produces-for(Pump1, Vibration1, Dampener1)

Thus we can conclude:
        receives(Actuator1, Power1)
        etc.
```

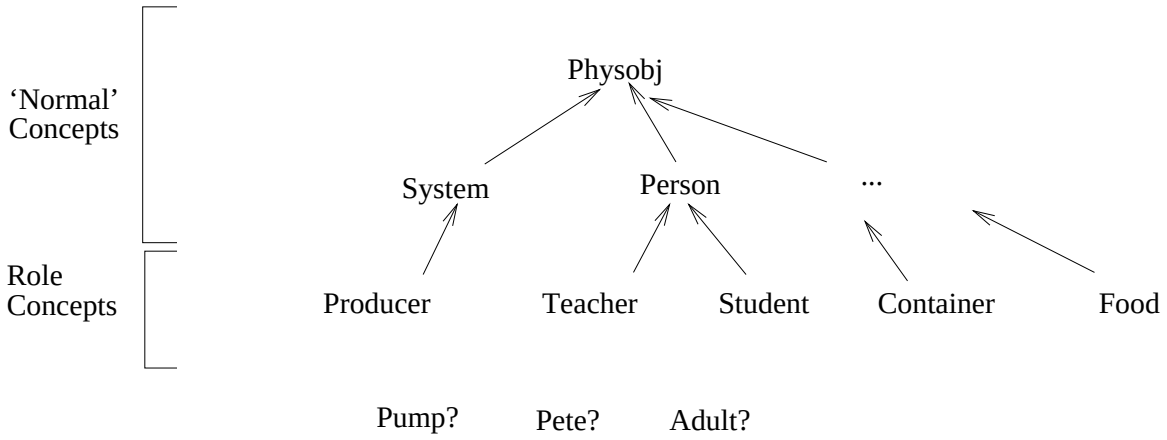
Figure 1: Formulation of Production: Reifying roles as concepts.

```
;;; -----------------------------------------------------------------------
;;;             MODEL OF PRODUCTION
;;; -----------------------------------------------------------------------
;;; "If a producer is producing a product for a consumer,
;;;  Then the producer supplies the product to the consumer, and sends
;;;  the product, and the consumer recieves it."
Forall Pdn:Production
        Forall P:Device, C:Device, R:Entity
        producer(Pdn,P) & consumer(Pdn,C) & product(Pdn,R)
                        -> supplies(P,R,C) & sends(P,R) & receives(C,R).

;;; -----------------------------------------------------------------------
;;;             USE OF THAT MODEL FOR A HYDRAULIC SYSTEM
;;; -----------------------------------------------------------------------
Instances are Pump1, Actuator1, Dampener1, etc. We place these in the
taxonomy as follows:
        Pump1:Device
        Actuator1:Device
        Dampener1:Device
        Power1:Entity
        Vibrations1:Entity
        Production1:Production      ; There are two instances of production
        Production2:Production      ; in the hydraulic system....

Then describe the production instances:
        producer(Production1,Pump1)
        consumer(Production1,Actuator1)
        product(Production1,Power1)

        producer(Production2,Pump1)
        consumer(Production2,Dampener1)
        product(Production2,Vibrations1)

Thus we can again conclude:
        receives(Actuator1, Power1)
        etc.
```

Figure 2: Formulation of Production: Describing roles as relations between objects and models used to describe those objects.

3 Discussion

There are several important points to note. First, both formulations are valid, and equivalent in what they say. So, from this point of view, the issue becomes more stylistic rather than fundamental. However, the first representation (Figure 1) has some troubling aspects: We might end up with a taxonomy looking like:



and then start wondering "What sort of things can be producers?" [ie. "What is a producer?"]. etc, a source of never-ending debate in trying to sort out how to place role concepts in the taxonomy. Instead, perhaps a better approach is to ask first "What is our model of production?" and then "Where can our model of production be applied?". In addition, looking at Figure 1, the role concepts of `Producer` etc. appear to be vaccuous – the properties which are conferred on instances of `Producer` are based on the `produces-for` relation and its implications, *not* on their membership of the class `Producer` (they do not inherit anything). We could just as well have written in the model of production in Figure 1:

```
Forall P:Device, R:Device, C:Entity
        produces-for(P,R,C) -> supplies(P,R,C) & sends(P,R) & receives(C,R).
```

and not mentioned the concepts `Producer`, `Consumer` at all! Rather, the terms 'producer' etc. are perhaps better thought of as (meta-theoretical) labels for the arguments in the `produces-for` relation[2]. In fact, Figure 2 simply reifies occurences of the `produces-for` relation as an object, and introduces relations `producer` etc. to relate that object to its arguments, a standard representational transformation.

The main conclusion from all of this is that, whichever representational approach is used, that the knowledge engineer should not waste a lot of time wondering "Is a X a type of `Producer`?" (say); rather, the first question should be "What would a model of production look like?", and then "Where are there instantiations of it in the real world?". The emphasis should be on axiomatizing the models, not building the isa hierarchy.

[2]The language LIFE allows terms to be (optionally) annotated with labels for their arguments, if desired, eg. `produces-for(producer=>P,product=>R,consumer=>R)`