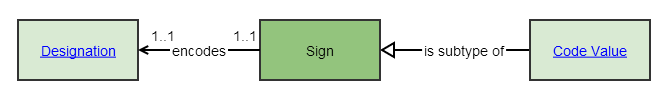
# GSIM “sign”



## QUESTION from group:

We are not sure we understand the purpose of the object “sign” and there seems to be some uncertainty about it in the GSIM world also.

## RESPONSE from Dan Gillman:

Sign seems to be very confusing to people for reasons I don’t quite understand. The best way to illustrate how signs work is to give the following analogy test:

“pain” is to “rue”, as “bread” is to what?

The answer is “street”. This may be obvious to you as an international traveler, but the STRING “pain” forms the French word for bread and the STRING “rue” forms the French word for street. As English speakers, we typically interpret “pain” and “rue” as English words with entirely different meanings. But, we make an interpretation.

Now, sign is a more general idea than string, but in computing, strings are by far the most common example. Traffic lights, traffic signs, and warning labels on packages are all signs. In use, they are designations, because there is meaning behind them. But, think about a factory where STOP signs are made. There is no idea of stopping associated with their manufacture. Someone cuts an octagonal metal plate, paints it red with an off-white narrow border, then stencils in the letters “S”, “T’, “O”, and “P”. When this octagonal metal plate is fastened to a pole by a traffic intersection, it gains meaning.

The main consequence of this is that in terminology the string we write and the meaning associated with it are independent of each other. The string is a sign, the meaning is a concept. A word or term or code is a sign associated with a concept which denotes it. Terms, codes, and other symbols are called designations. Words are like designations. When we interpret a word or designation, we infer the meaning (concept) associated with it. If we are looking at a sign only, there is no interpretation possible. Codes typically look this way unless we have the mapping (association) in front of us. Terms are linguistic and look like words, so their meaning is easier to infer, though often not completely. Think “unemployment” as defined by BLS.

In your attached model, code is shown as a sub-type of sign. Chris insisted on doing this when modeling GSIM. I reluctantly agreed, but that was an example of how the modeling skews the meaning and usage of things we define. This would mean a code is a kind of string with certain rules for its formation. I’d rather see it as a designation, chiefly used in representing categories in classification systems in statistics.

I think in statistical offices, codes present major problems to people. Codes are used in special ways, and as a result, they are given a special importance. Because of this, it is hard to see how they behave similarly to other designations. However, the usage of codes versus other terms (such as short and long labels in classification schemes) does not change what codes are – signs associated with concepts.

Another issue is synonymy and homonymy. Two codes can designate the same concept (synonyms) – think of code changes for the same category across versions of a classification. Also, the same code can be used to designate categories in several classification schemes (homonyms). For instance, the code “M” can be used to designate the male sex, male gender, married marital status, Monday (in English!), and many other obscure possibilities.

### REPLY from Wendy:

For me the big thing is that this is an abstraction layer. We are and will be dealing with other forms of codification. Larry brought up the example of capturing images of American Sign Language. I think we may need the abstraction as something to attach future possible code values to that may be something other than xs:string.