**CNO Documentation**

Yann Le Franc, PhD

January 2013

Welcome to the Computational Neuroscience Ontology documentation.

Introduction

Why CNO?

With the increased number of computational neuroscience studies comes the problem of reproducibility and reusability of either the whole or part of a model. The diversity of model implementations makes it unlikely that one laboratory can easily reproduce the results obtained by another group, even if the model is deposited in an openly accessible database. To solve this problem, the community needs to define and adopt standards for model description that will make model sharing more efficient. These standards should propose a common format of representation, independent of any given simulator that could be imported or exported by the different simulators.

To address this problem, standardized languages have been developed by and for the community, such as NeuroML (), PyNN () and NineML (http://software.incf.org/software/nineml). Although these languages enable software interoperability and therefore model reuse and reproducibility, they lack semantic information that would facilitate efficient model sharing and retrieval.

The creation of such XML language, as NineML, enables semantic web technologies. These recent technological developments, thought to trigger the next World Wide Web (web3.0), aim at providing “meaning” in the web content that will usable by software (i.e., machine-readable). This addition of “semantic” information would then allow a better organization of the data on the web and provide the basis for their optimal querying.

This “semantic” information is contained in ontologies. Ontologies are formal models of knowledge in a particular domain, and are composed of classes that represent concepts defining the field as well as logical relations that link these concepts together.

The development of ontologies has been the subject of extensive research in the biomedical sciences, with the development of well-known ontologies such as the Gene Ontology (GO), System Biology Ontology (SBO), NIFSTD Neuroscience ontologies, the Phenotypic Quality ontology (PATO) and many others. These machine-interpretable solutions, using the mathematical framework of first-order logic, allow computers to make inferences on datasets and to reveal particular relations between different types of data, or simply to provide a more intuitive and efficient information retrieval for domain-experts.

Despite the large number of existing Biomedical ontologies (e.g., 304 referenced by the NCBO bioportal http://bioportal.bioontology.org/), only few ontologies are being developed to represent biological modeling, but most of them do not have yet a substantial overlap to be used for Computational Neuroscience. It is then clear that an ontology describing the field of Computational Neurosciences would represent a major tool for the international modeling community.

What is CNO?

In the context of the INCF Multi-Scale Modeling (MSM) program, we have developed an ontology to annotate Computational Neuroscience models described with NineML and other structured model description languages.

CNO is a specific controlled vocabulary composed of classes that representing general concepts related to computational neuroscience and organized in a hierarchy of concepts as well as the logical relations that link these concepts together.

An alpha version of CNO is currently accessible on bioportal, the biomedical ontology portal provided by NCBO (<http://bioportal.bioontology.org/ontologies/3003>).

How has it been developed?

The first step was to gather the important terms or concepts from the literature and more precisely from textbooks (REFERENCES) and build up a preliminary list based on the common knowledge from the MSM Task Force.

Based on this list of terms, we developed a first version of the Computational Neuroscience Ontology (CNO), based on the OWL-DL standard proposed by W3C (http://www.w3.org/TR/owl-features/). This particular form of the standard enables the use of inferences.

The design of CNO follows some of the recommendations of the Open Biological and Biomedical Ontologies (OBO) community and it is compatible with the ontologies developed and maintained within the NIF. Integration within this large federation of neuroscience ontologies has two main advantages: i. it allows the linking of models with biological information, creating a bridge between computational and experimental knowledge bases; ii. as ontology development is an iterative process that relies on inputs from the community, NIF has developed NeuroLex (http://neurolex.org), an effective collaborative platform, available for community inputs on the content in CNO.

CNO classes and relations have unique identifiers that allow unambiguous annotation of digital resources such as web pages or model source code. These unique identifiers are of the form cno\_XXXXXXX (7digits) that allow to create unique URI that references this particular class

The information regarding the meaning is added to the classes with the use of 1- a label, which will contains the human-readable name, 2- a human-readable definition that explicitly states the parent class and the property that characterize the class from the different other members of the parent class e.g.

To add the semantic meaning within the ontology, we uses AnnotationProperties that represent the metadata attached to the URI such as label, definition, synonym, definingCitationURI. These terms come from different annotation ontology such the Dublin Core, SKOS, OBO-annotations

For instance, Has main source of information for definitions we used Scholarpedia, Wikipedia, reference articles. When available links to the resource used to create the definition is described in the field “definingCitationURI” or “definingCitationId”

A concrete example being often better than long explanations, here is one just for you:

The class describing the Leaky Integrate-and-Fire model

URI: <http://purl.org/incf/ontology/Computational_Neurosciences/cno_alpha.owl#cno_0000066>

Label: leaky integrate-and-fire

Definition: a defined model that is an integrate-and-fire model with a leak component added to the membrane potential, reflecting the diffusion of ions that occurs through the membrane when some equilibrium is not reached in the cell.

definingCitationURI: <http://en.wikipedia.org/wiki/Biological_neuron_model#Leaky_integrate-and-fire>

CNO has been developed using the ontology editor Protégé (version 3.4.4 build 579) for the version up to CNO v0.2.4.. For the development of CNO version 0.5, we use Protégé version 4.1.0 build 239.

CNO is currently (January 2013) composed of 215 classes, representing general concepts related to computational neuroscience and organized in a hierarchy of concepts.

We are currently collaborating with NIF ontologists and with the OBO community, to integrate CNO into NeuroLex, to provide a user-friendly interface for the community.

The difficulty of designing such ontology: a rationale for the contribution of the community.

Problems with the terminology. Need to clearly define concepts.

Scope of CNO

CNO aims at providing a controlled vocabulary to annotate Computational Neurosciences resources such ModelDB, Scholarpedia.

CNO can be used to annotate any type of resource: web pages, wiki pages, database entries, model XML descriptions, source code, …

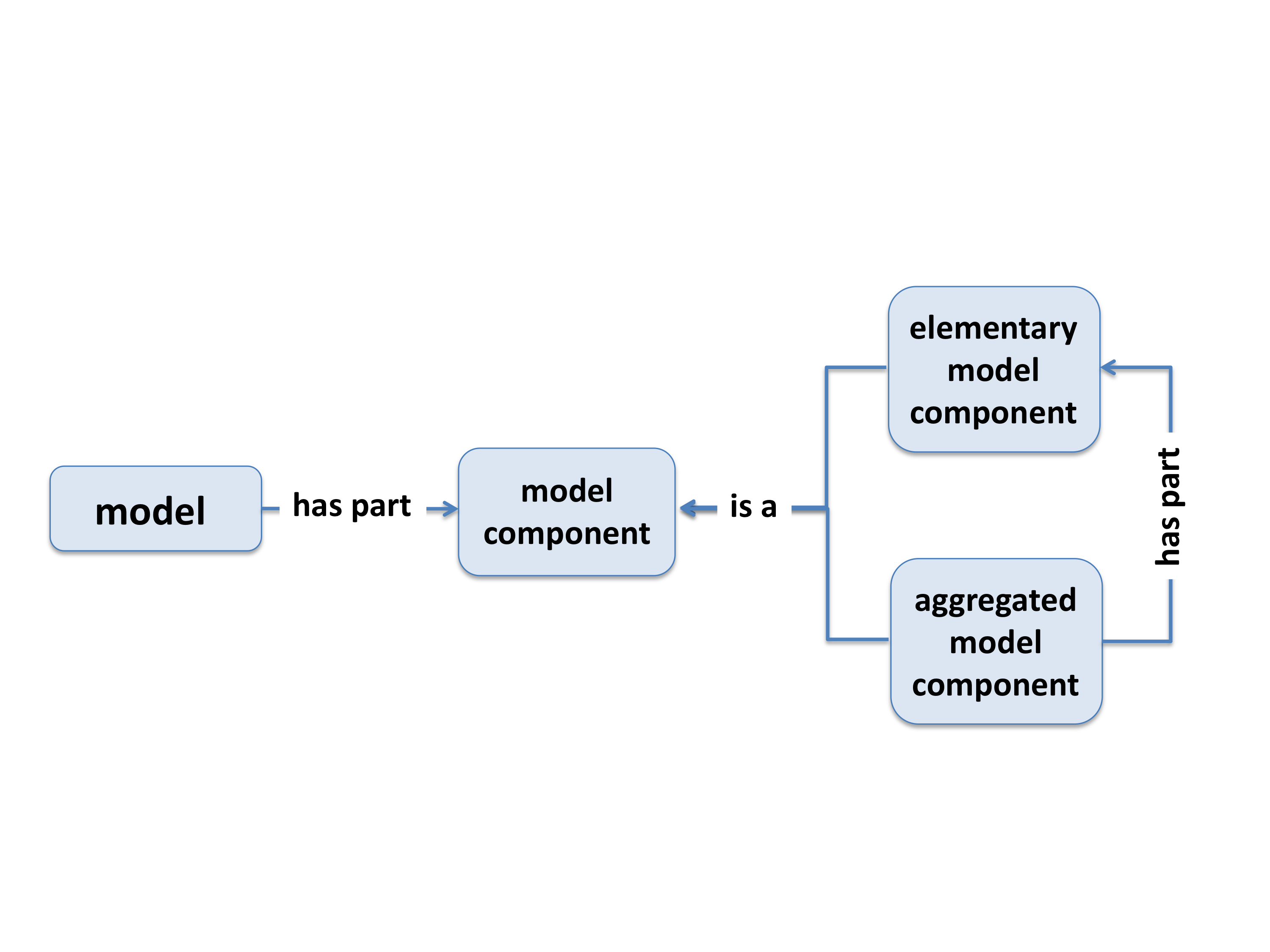
Such process would allow to interconnect the different types of information together providing the possibility of proposing valuable resources.

As Computational Neurosciences is tightly bounded to Neurosciences, the integration of CNO into neurosciences ontologies should be a priority.

CNO internal model

To represent the large diversity of models, we considered that a given **model** is the results of the aggregation of different building blocks called **model component** (figure 1). These components should be general enough to allow the construction of any type of models de novo or based on previous models. This allows us to represent the typical case where a new model is build from an existing model with the addition of a new component or the change of a particular component (e.g. using a different cell model, adding a different type of synapse models or adding a model of a particular ionic conductance).

The figure 1 below shows the classes in CNO that were created to implement this particular model.



The **model** class is an upper class that represents all the models that could be encountered. Any model that will be annotated is then an instance of this class.

The **model component** class is also an upper class representing the set of components that can be used to build a model. It can be further subdivided into **elementary model component** and **aggregated model component**.

The **elementary model component** class includes the smallest components that can be used in models e.g. voltage, current, indices, kinetic rates, amplitude, …

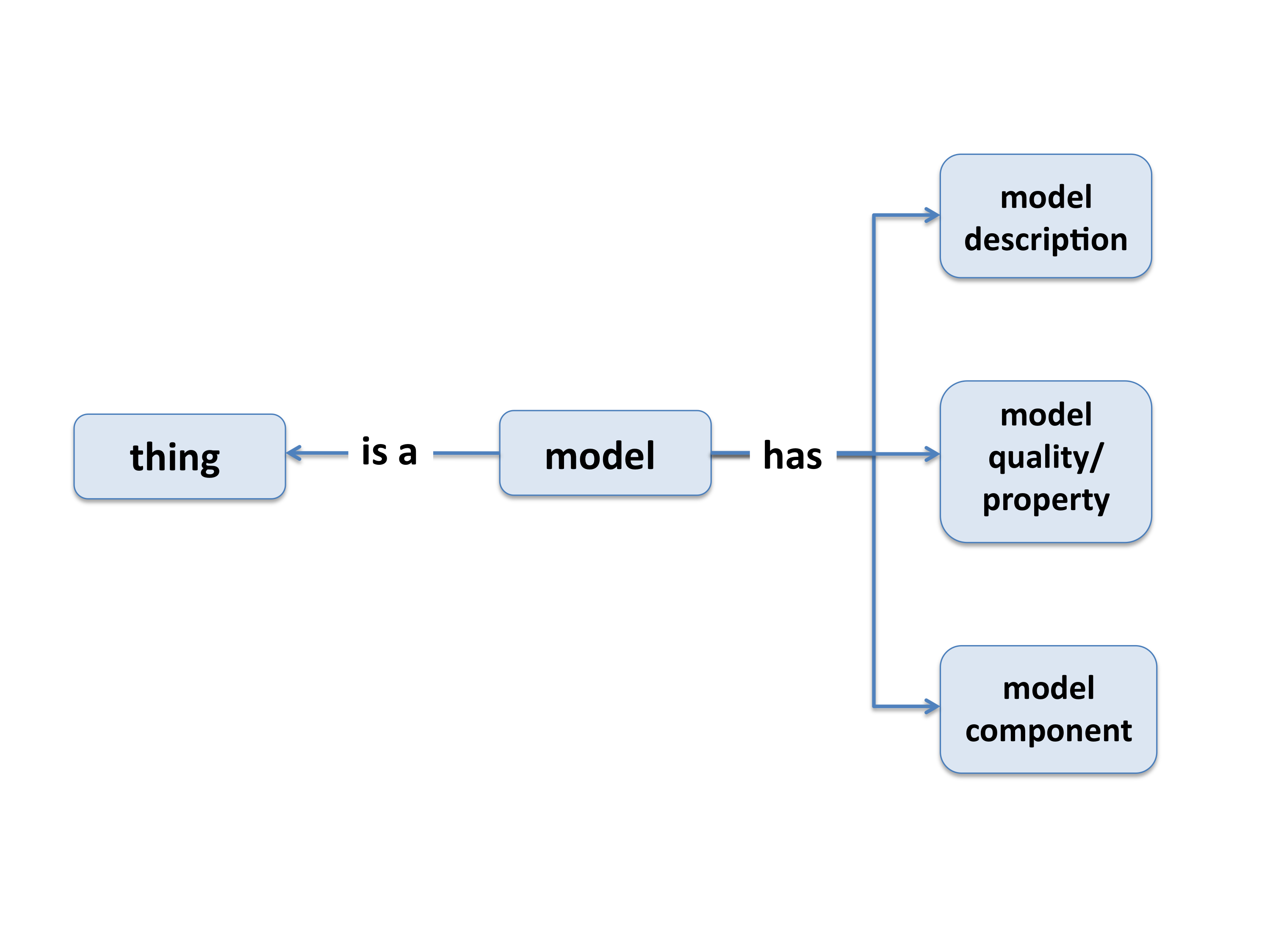
The terms contained in this particular class have been for a large part imported from SBO. The procedure used to import these terms into CNO is described later (see XXXX). These elementary components can be used as variable or as a parameter in the model.

The **aggregated model component** class represents more complex components such as ionic current model, morphology, network layout which result themselves from the aggregation of elementary components and mathematical operators or functions. It is important to mention that the scope of this ontology is not supposed to cover such mathematical functions and operators. The aggregated components are classified depending on the level of description (cell, synapse, network, synaptic plasticity).

The link between the model and its component is made through the ObjectProperty “*has part*” defined in the Relation Ontology (link).

This simple model seems to be efficient to represent any model structure. However a model has also some other important characteristics such as its description (publication, XML file, ModelDB entry, …) or functional such as bursting, oscillation, synchronization that emerging from the model structure.

To account for these important notions we added two upper classes as shown in figure 2.



The **model description** class represents the different possible description for the model: publication, a particular implementation, a XML representation, ...

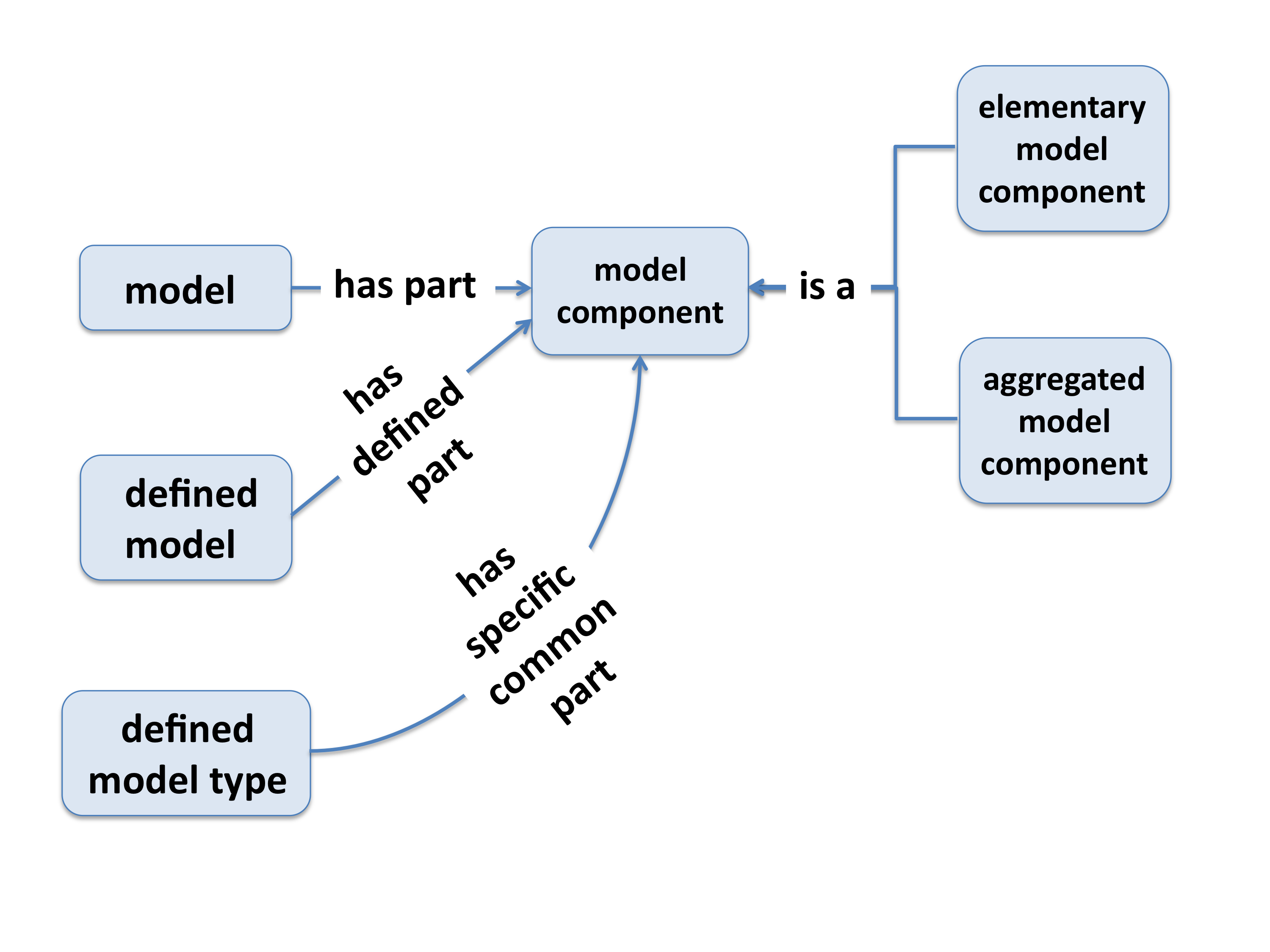
As this class is rather broad, we only considered the PMID and the ModelDB accession number.

The **model quality**/property class represents the properties that emerge from the particular structure of the model. This class is rather complex and broad, we chose to focus on populating it in later versions of CNO. To have an example of the type of concept that should be included, you can look at the list proposed on ModelDB (<http://senselab.med.yale.edu/ModelDB/FindByConcept.asp>).

Finally, we created two additional upper classes: **defined model** and **defined model type** (figure 3).

The **defined class** represents the set of models that we could consider as “standard” models e.g. Leaky Integrate-And-Fire, Izhikevitch model, Markram and Tosdyks model, Fitzhugh Nagumo, Morris-Lecar, Hodgkin and Huxley and many more. These different models have well known characteristics and are often used as a base for building up different models.

The **defined model type** is a classification of the different models. It represents the different general types of models such as artificial neural network, spiking network, point process, …



The **defined model** class is linked to the **model component** class using the relation *“has part”*. These relations are refined with restrictions to express that one particular model has some defined components.

To give an example, the current definition of a Leaky Integrate-and-Fire describes the model with the following components: a fixed spiking threshold, a stimulation current or a synaptic current, a point morphology, a refractory period and a leak current.

Using this model, it would become possible to classify any annotated model that present the particular features as a Leaky Integrate-and-Fire model for instance even if this information hasn’t been provided in the annotations.

We are planning to create these restrictions once the ontology is more mature.

Work done on CNO version 0.5:

The main objective of this update of CNO is to propose an ontology that 1- could be integrated into the Neuroscience ontologies developed by the Neuroscience Information Framework (NIFSTD ontologies) and interoperable with other Biomedical Ontologies and 2- includes terms from existing ontologies such as the System Biology Ontology (SBO), the Ontology for Biomedical Investigation (OBI) or the Information Artifact Ontology (IAO).

The first step of the work was to reconsider all the definitions proposed in CNO version 0.2.4 and add most of the missing definitions. The proposed definition are structured following the recommendations of the OBO foundry community where the first part of the definition relates the term to its superclass (genus) followed by the propriety or quality that differentiates entities belonging to this particular class compared to the other entities of the superclass (qualia).

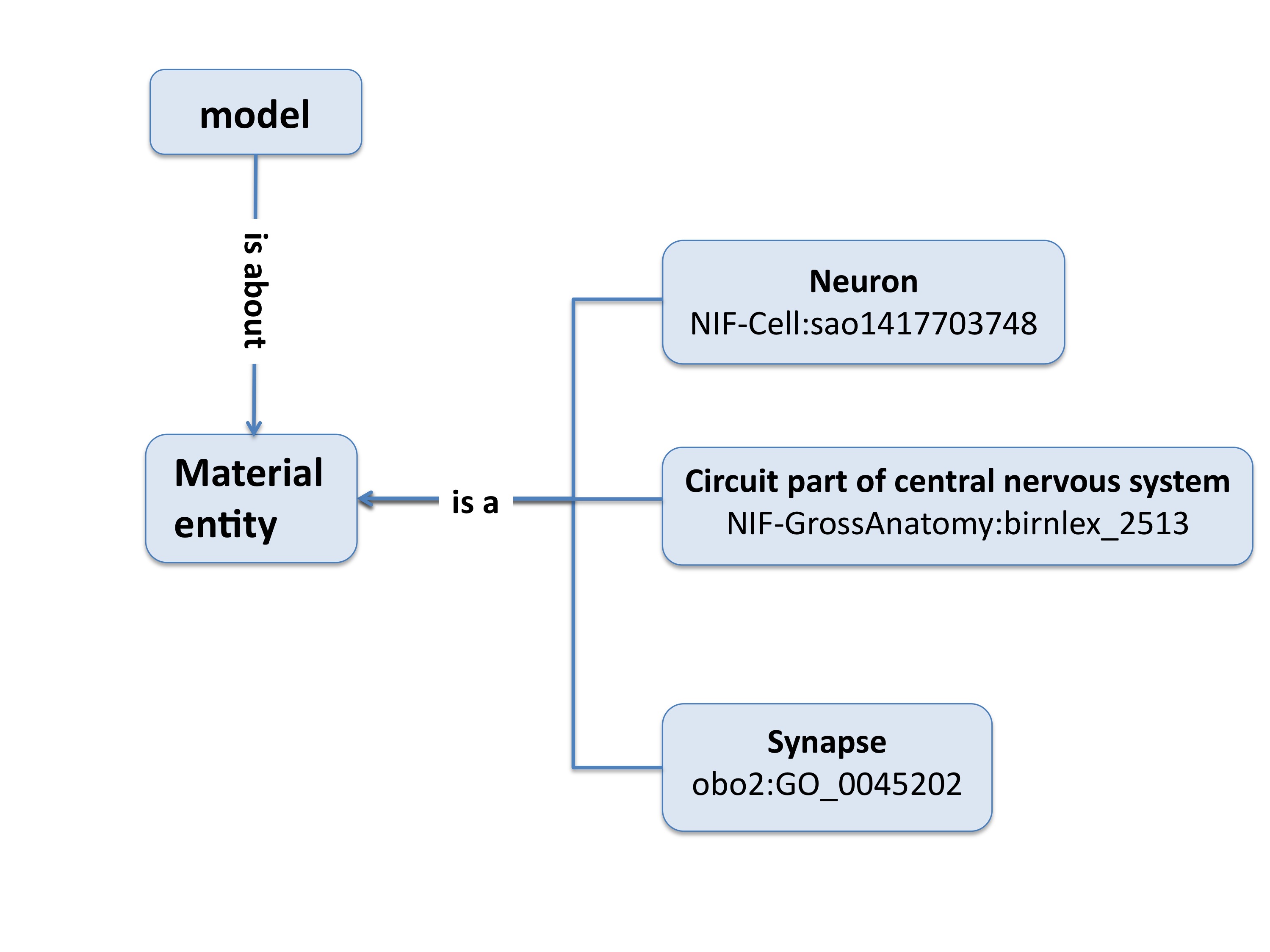
The second step was to integrate CNO classes into the Basic Formal Ontology superclasses.

The third step was to import terms that are defined in other ontologies. For this, we used the MIREOT format and the Ontofox service.

Integration into the Basic Formal Ontology

The Basic Formal Ontology (BFO) is an upper-level ontology that describes material entities bounded in space. BFO is currently used in several biomedical ontologies including the NIFSTD ontologies.

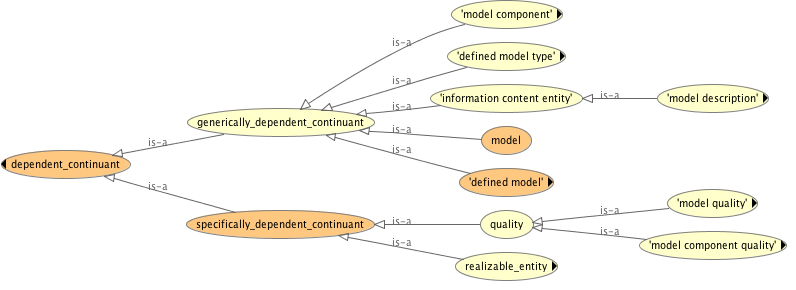
Although a model can be considered as a material entity through it implementation or the publication that describes it, models cannot really be bounded in space.



However, models aim to describe material entities (real neurons, synapses, or part of the brain). IAO has worked on a similar problem to deal with IT content.

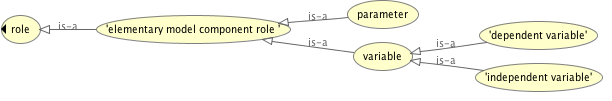
Based on this work, our assumption for the integration into BFO is to consider that a **model** “is about” a **material entity**. The relation “is about” is proposed by IAO and provide a link between a **material entity** named an **information bearer** and **generically dependent continuant** named **information content entity**.

We integrated CNO within a similar model.



To allow for a particular elementary model component to be either a variable or a parameter depending on the model (e.g. the membrane voltage could be a variable in the subthreshold equation of an IAF or could be a parameter in some other model.)

To address this issue, we used the structure of BFO to define that being a parameter or a variable is a role that is attributed to a particular elementary model component.



Why do we need an ontology for?

The development of such ontology, in the context of the INCF Multi-Scale modeling program, will play a tremendous role for the federation of both Computational Neurosciences resources, such as Scholarpedia (http://www.scholarpedia.org), ModelDB (<http://senselab.med.yale.edu/modeldb/>), and of neurosciences resources, referenced under the umbrella of the Neuroscience Information Framework (NIF, http://www.neuinfo.org/). This will allow creating an efficient platform to design multi-scale models of the brain based on multi-scale experimental datasets.

The answer to this question requires using YOUR imagination.

So let’s dream for a minute that we are in a perfect world…

Imagine a perfect sunny day. You woke up this morning with an idea of model that you believe is the best model ever to describe what you are interested in (basically our every day routine…).

Being cautious before asserting that it is the best idea ever, you want to check that no similar model already exists, how you could implement your idea and which models you could reuse to test your idea… so you are opening your favorite browser and then start thinking about the resources to use for

1. retrieving shared model source code with ModelDB (ref)
2. getting immediately theoretical knowledge about the model (Scholarpedia)
3. reusing models with standardized languages, such as NeuroML (), PyNN () and NineML (<http://software.incf.org/software/nineml>), which allow for software interoperability and are being developed by and for the community (<http://software.incf.org/software/nineml>)

And then the dream stops when you start thinking about getting a precise answer to questions like:

“What are the connectivity rules that implement a feedforward neural network model?”, “What are the different ways to implement this rule?”, “What are different platform on which this rule has been implemented?”, “What are the parameters ranges used with these rules?”, “What are the models developed in NEURON that implement a model of a particular calcium channel using Markov chains to represent the gating of the channel?”, “What are all the models of cell of interest that have a non-linear distribution of the conductance along the dendritic trunk?”.

Although it is still possible to get an answer to these questions using the different existing resources, getting a focused and synthetic answer to these highly domain-specific questions is still not possible due to the lack of semantic information to describe the different type of data.

We are proposing here a solution based on semantic web technologies and semantic annotations. requires developing a common Computational Neurosciences vocabulary that will be used for annotating the different resources.

To make this dream continue you need to be able to query available resources based on specific Computational Neurosciences terms (e.g Markov chain, Hodgkin and Huxley formalism, connectivity rule, and so on and so forth…).

For this, the different available resources we mentioned earlier must be annotated with semantic information specific to Computational Neuroscience. This semantic information should be in a machine-readable format.

and called an ontology. This approach, based on semantic web technology used by the Neuroscience Information Framework to propose a data federation that is accessible via a web interface.

However, tremendously valuable resources currently exists to make your life easier by providing support to:

Now that you are thinking about it for a minute, I am guessing that you are now wondering if it wouldn’t be indeed great if we could continue the dream and query the web with questions like:

… and get of course relevant focused answers you are looking for…

To make this practical dream come true we need

Currently only parts of these questions can be answered on different valuable resources for the community like ModelDB which actually provides a great search interface centered around neuroscience-specific concepts (cell type, channel type, …) or Scholarpedia which aggregates articles on different aspects of Computational Neurosciences.

This ontology can be a simple list of terms organized in hierarchical manner or can contain a knowledge model to represent the inter-relations between the different terms that will allow to make inferences and ask such complex questions.

For this purpose, this vocabulary needs to be implemented in a machine-readable format on which reasoning agent will operate first-order logic operations.

The diversity of modeling approaches in computational neuroscience makes model sharing, retrieval, reuse and reproducibility difficult and even sometimes impossible.

To address the problem of model reproducibility and reuse, standardized languages such as NeuroML [1], PyNN [2] and NineML (<http://software.incf.org/software/nineml>), have been developed by and for the community. Although these languages enable software interoperability and therefore model reuse and reproducibility, they lack semantic information that would facilitate efficient model sharing and retrieval.

1. retrieving shared model source code with ModelDB (ref)
2. getting immediately theoretical knowledge about the model (Scholarpedia)
3. reusing models with standardized languages, such as NeuroML (), PyNN () and NineML (<http://software.incf.org/software/nineml>), which allow for software interoperability and are being developed by and for the community (<http://software.incf.org/software/nineml>)

In the context of the INCF Multi-Scale Modeling (MSM) program, we have developed an ontology to annotate spiking network models described with NineML and other structured model description languages. Ontologies are formal models of knowledge in a particular domain and composed of classes that represent concepts defining the field as well as the logical relations that link these concepts together [3]. These classes and relations have unique identifiers and definitions that allow unambiguous annotation of digital resources such as web pages or model source code. Implemented in a machine-readable format, these knowledge models can be used to design more efficient and intuitive information retrieval systems for experts in the field.

What is CNO?

CNO is a domain specific terminology that can be used for annotating resources in Computational Neurosciences.

It has been developed using the cutting-edge technology to allow the construction of a logical model that would enable inferences.