

# CATMAID data model

Stephan Gerhard, Stephan Saalfeld and Mark Longair <[longair@ini.phys.ethz.ch](mailto:longair@ini.phys.ethz.ch)>

**CATMAID**, the Collaborative Annotation Toolkit for Massive Amounts of Image Data, is a web-based platform for the annotation of very large data sets, such as those produced by serial section transmission electron microscopy. The aim of the CATMAID data model is to describe annotations of image data in terms of Subject-Relation-Object triples suitable for mapping to semantic web technologies and incorporation into controlled vocabularies such as those defined by the Open Biomedical Ontologies group. Our other aim is to be able to store these logical relationships efficiently in PostgreSQL. The following sections describe our current model and the plans for possible changes.

## Current Entities and Relationships

The highest level distinction between the entities we are modelling is between **locations** and **class\_instances**. The distinction between these is that a **location** is a physically located point in space, whereas a **class\_instance** is an abstract concept, only linked to a point in space via relationships with other entities.

There are two subclasses of **location**: **treenode**, which represents a node as part of a tree structure, and **connector**, which is used to link locations in a many-to-many fashion, typically used for annotating synapses.

Examples of **class\_instances** include neurons, groups of neurons, groups of groups, skeletons, synapses and pre- and postsynaptic terminals.

The current set of relationships which are allowed between these entities are:

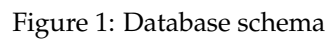
<i>subject</i>	<i>relation</i>	<i>object</i>
<i>treenode</i>	<b>labeled_as</b>	class_instance
treenode	<b>element_of</b>	class_instance
treenode	<b>model_of</b>	class_instance
connector	<b>model_of</b>	class_instance
class_instance	<b>model_of</b>	class_instance
class_instance	<b>part_of</b>	class_instance
class_instance	<b>is_a</b>	class_instance
class_instance	<b>postsynaptic_to</b>	class_instance
class_instance	<b>presynaptic_to</b>	class_instance

These relationships are all asymmetric. Every **class\_instance** is an instance of a particular **class** - at the moment, the defined **classes** are **skeleton**, **neuron**, **group**, **label**, **root**, **soma**, **synapse**, **presynaptic terminal**, **postsynaptic terminal**.

Examples of the uses of each of these relationships are given below:

### **labeled\_as**

During annotation, it is frequently useful to annotate a point in the tree with a text-based tag, which might be "TODO", "possible synapse here", etc. This relationship links a **treenode** to a label of this kind.



### **postsynaptic\_to, presynaptic\_to**

We are representing the polyadic synapses found in our data with a connector placed over the synaptic cleft. A connector is a `model_of` a synapse. Treenodes that are pre-synaptic and post-synaptic partners are in a `model_of` relationship with `presynapticterminal` and `postsynapticterminal`. The `postsynaptic_to` and `presynaptic_to` relationships are then used to link the synaptic terminals to their synapse. These relationships could describe many-to-many relationships across synapses, although there is typically only a single presynaptic terminal in the relationship.

### **element\_of**

The `element_of` relationship is currently used to indicate that a treenode is part of a particular skeleton. This is distinct from the standard `part_of` relationship to indicate that the subject is a treenode rather than a `class_instance`.

### **model\_of**

The concrete annotations used in CATMAID should be tissue-type agnostic; in other words, they should not be specific to annotating neuronal structures. The `model_of` relationship links a basic annotation to the particular biological structure that it is representing. One example use of this is to create a link between a skeleton and the neuron it represents the midline of. The same neuron can be modelled by skeletons traced by different users. As another example, a connector is a model of a synapse.

### **part\_of**

This relationship indicates that one `class_instance` is part of another; for example, a soma is `part_of` a neuron or a presynaptic terminal is `part_of` a skeleton. This should be the same as the `part_of` relationship used in the OBO foundry<sup>1</sup> ontologies.

### **is\_a**

This relationship indicates that one `class_instance` is a subtype of another; for example, a neuron `is_a` cell. This should be the same as the `part_of` relationship used in the OBO foundry.

## **Representation in PostgreSQL**

The representation of these entities and relationships in PostgreSQL makes use of the database's inheritance feature. If one table inherits from another, then the descendant has all the columns of the former, and various queries enacted on the parent table will also include rows from all descendant tables. At the top level of this hierarchy in CATMAID is the **concept** table, which provides columns for the *user\_id*, *project\_id*, *creation\_time*, *modification\_time* and a unique *id*. All the other tables described below similarly inherit from **concept**.

Each possible relation is described at the top level by a row in the **relation** table. For example (with some columns elided):

---

<sup>1</sup><http://www.obofoundry.org/>

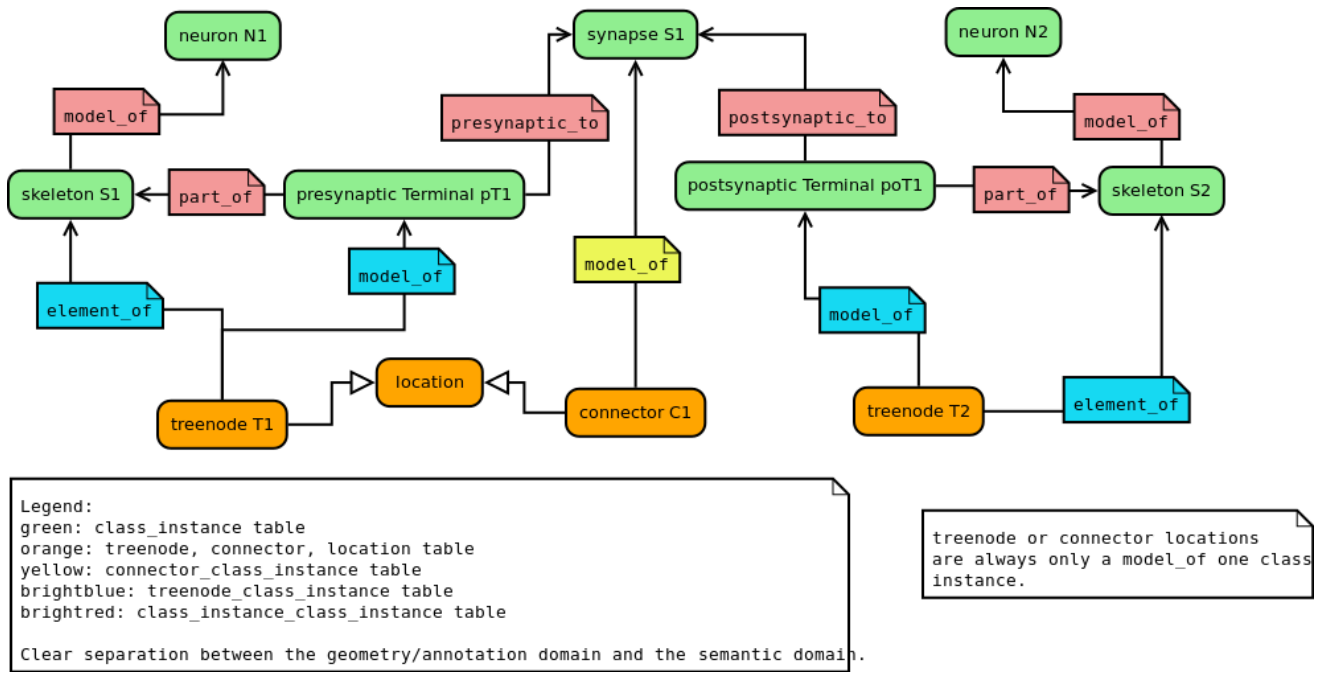


Figure 2: Treenode-connector relationships with the semantic annotation

<i>relation_name</i>	<i>isreciprocal</i>
labeled_as	f
postsynaptic_to	f
presynaptic_to	f
element_of	f
model_of	f
part_of	f
is_a	f

An instance of one of these relations is represented by a row in a table which inherits from the **relation\_instance** table. The **relation\_instance** table adds a *relation\_id* column to specify which relation this is an instance of. The three tables that inherit from **relation\_instance** are **class\_class**, **class\_instance\_class\_instance** and **treenode\_class\_instance**. The **class\_class** table describes the possible classes that can be linked by that particular relation. Which table is used for the concrete relationships depends on whether the subject of the relationship is a **class** or a **treenode**. The **treenode\_class\_instance** table has rows that link a treenode to a class instance, whereas the **class\_instance\_class\_instance** table links two **class\_instance\_class\_instances**.

Examples of classes from the **class** table are:

<i>class_name</i>	<i>showintree</i>
skeleton	t
neuron	t
group	t
label	f
root	f
soma	t
synapse	t
presynaptic terminal	t
postsynaptic terminal	t

The *showintree* column indicates whether this class should be shown in the tree display of the annotations in CATMAID's interface. Particular instances of each of these classes are represented by rows in the **class\_instance** table, which provide a name and id for each class instance.

Instances of the Subject-Relation-Object relationship between instances of two classes are represented by rows in the **class\_instance\_class\_instance** table:

<i>relation_id</i>	<i>class_instance_a</i>	<i>class_instance_b</i>
--------------------	-------------------------	-------------------------

Relations can also apply between a **class\_instance** and a **treenode**.

The **treenode** and **connector** tables inherit from the **location** table which expresses the location in space.

The distinction between a treenode and a location is that a treenode additionally has these features:

- *parent\_id*: A parent treenode, which may be null in the case of the root node in a skeleton.
- *radius*: a radius of the skeleton at that point.
- *confidence*: on a scale of 1 to 5, the confidence that the annotator has that the link to the parent node is correct. (5 is the most confident.)

These relationships between treenodes and class\_instances are stored as rows in the **treenode\_class\_instance** table:

<i>relation_id</i>	<i>treenode_id</i>	<i>class_instance_id</i>
11 [element_of]	240	561
11 [element_of]	241	561
11 [element_of]	242	561
...	...	...
23 [model_of]	268	2500
35 [labeled_as]	268	1562

('labeled\_as' gives a label to a particular treenode, 'model\_of' to mark the treenode as presynaptic terminal and 'element\_of' links a treenode to the skeleton it is a part of.)

Similarly, relations can apply between a **connector** and a **class\_instance**. Connector are points in space used to model the location of a synapse. Connector additionally has these features:

- *confidence*: on a scale of 1 to 5, the confidence that the annotator has that this connector actually is a synapse and links correctly. (5 is the most confident.)

These relationships between connectors and class\_instances are stored as rows in the **connector\_class\_instance** table:

<i>relation_id</i>	<i>location_id</i>	<i>class_instance_id</i>
23 [model_of]	300	1800
23 [model_of]	303	4800
...	...	...
35 [labeled_as]	268	15623

## File Format

For this web-based, database-backed system a concrete file format for import and export is not a critical as provision of an API for interrogating the database. However, we will initially use **TrakEM2**'s XML-based format in order to import data.