**Proposal**:

A ‘lego prototyping group’ will generate sample annotations that will drive future requirements and software development, and test the model, iron out modeling decisions, etc. The prototyper can either encode directly in Protégé as instance-level relationships (triples), or they can summarize as a diagram which another prototyper can encode in Protégé. The resulting OWL files will be saved in SVN.

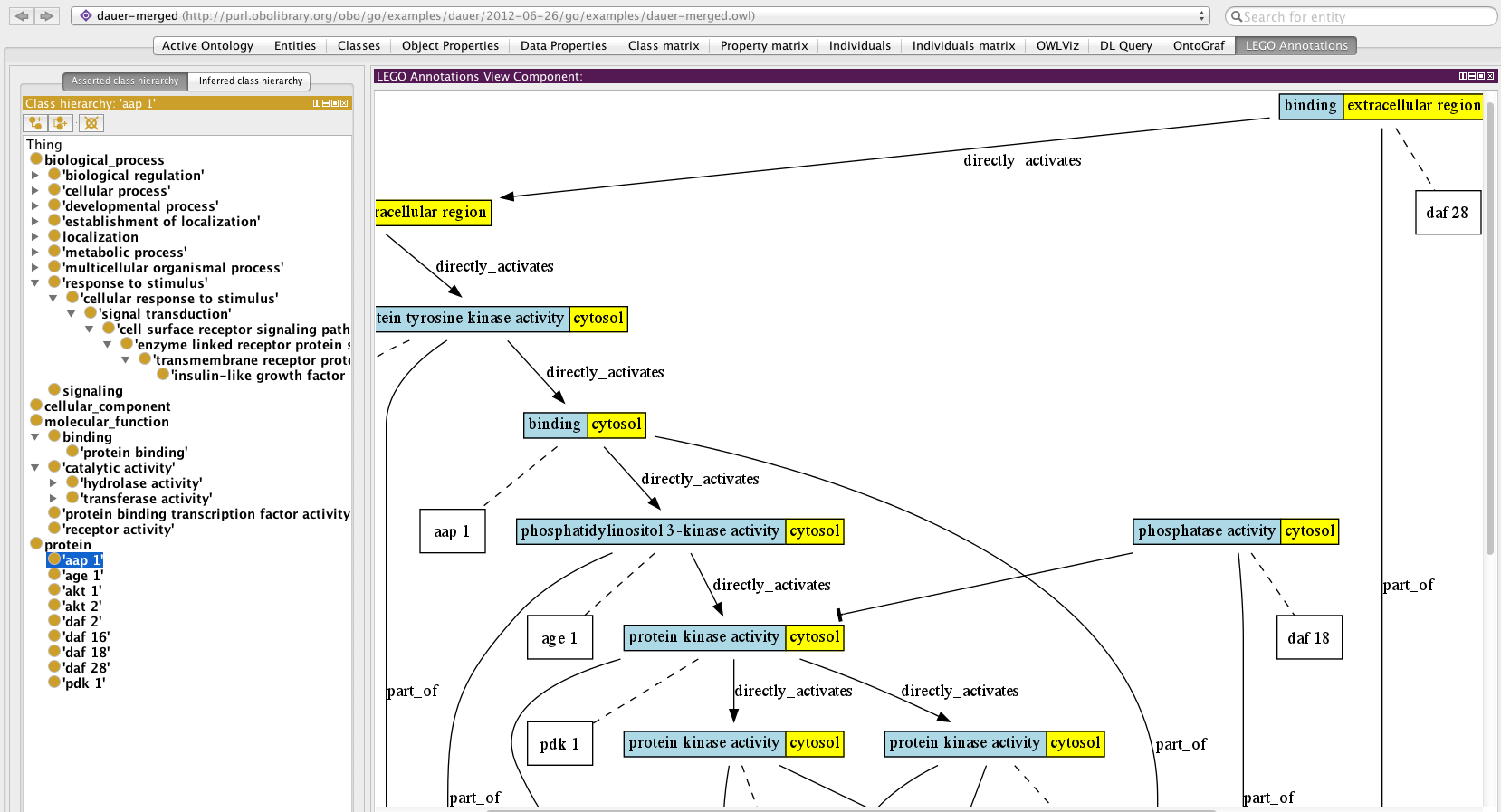
We have developed a simple plugin that visualizes the triples based on the ‘annoton’ model.

**LEGO plugin**:

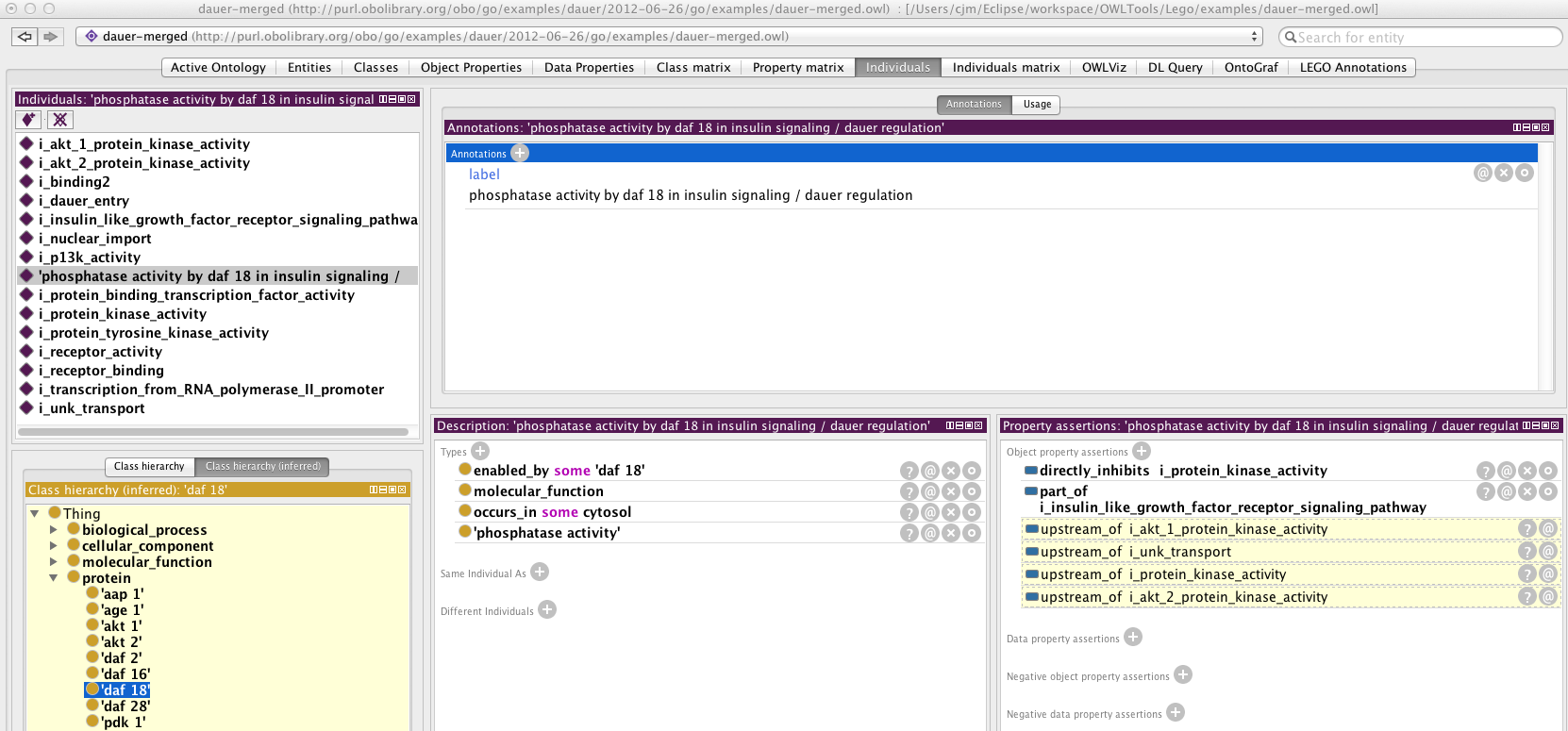
* Install Protégé 4.2
* Install the org.geneontology.lego plugin[[1]](#footnote-1)

Examples are currently in the owltools repo[[2]](#footnote-2) for software testing purposes, but can be moved to GO SVN. Recommended example: <http://owltools.googlecode.com/svn/trunk/Lego/examples/dauer-merged.owl>

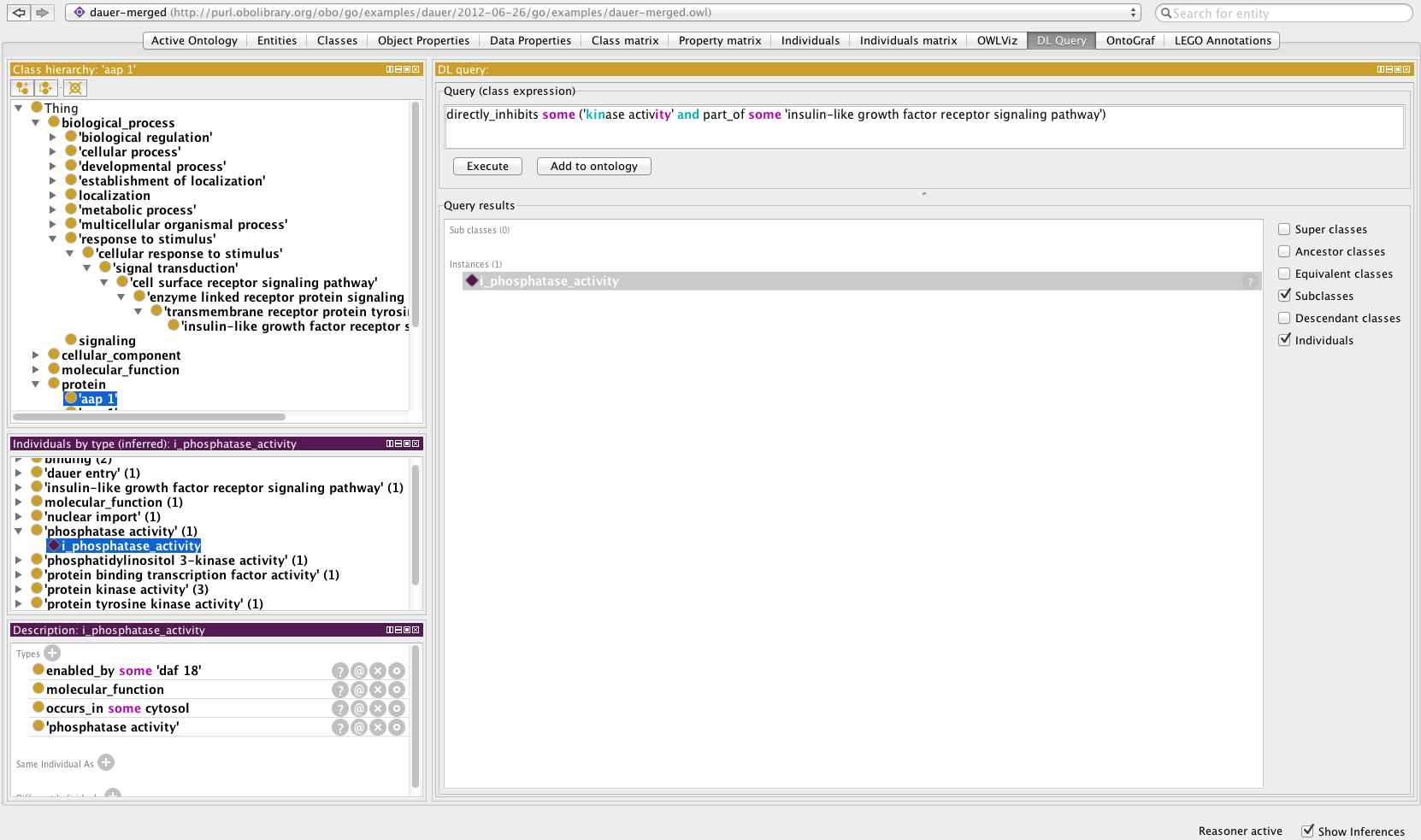
Once the example is loader, start HermiT, then choose Views/LEGO plugin. The top bar must be selected for the display to become active. The plugin finds all instances of MFs and displays their connections using a pre-defined set of relations.



*Figure 1: screenshot of LEGO annotation plugin with dauer example. Note the display is entirely driven by the underlying annotations encoded as triples*.



*Figure 2: Generic low-level OWL view of underlying assertions that drive display for a particular instance of an MF (a phosphatase activity that performs an inhibitory role during the dauer/insulin pathway). MF to gene product links use the enabled\_by relation. The location of the activity uses occurs\_in (already used in GO) to draw locations as yellow boxes in the LEGO plugin. Reasoner inferences in pale yellow boxes on lower right.*



*Figure: DL query for functions that inhibit other functions in the insulin pathway. One instance ‘i\_phosphatase\_activity’ is returned (a bit confusing in this example. Each function instance / annoton should have a meaningless numeric ID and a label such as ‘phosphatase activity by daf-18 in insulin pathway during dauer arrest’). Note that this is just the generic Protégé query tab, not the LEGO plugin*.

**Instructions for making a LEGO annotation set**:

These instructions are for the prototypers – lessons learner here will drive the development of the actual tool curators would use.

(instructions very sparse – to be filled in later. It is unlikely you’ll make much headway here unless you attended the Hinxton course)

Create a new ontology (each annotation bundle is treated as a separate ontology == file), name it according to specified guidelines (TBD). Add an import to GO, or a GO slim. Import the protein set for the organism(s) of interest (instructions to follow). Add an import to the lego relations subset of RO (todo).

Make sure you have set up ID generation. Each MF instance (‘annaton’) will get a unique ID. Don’t worry too much about these as this is just test data for now.

Start by choosing the first gene product in the pathway. What function is it executing? Create a new individual (individuals tab) and under ‘Types’ put the MF. You can give it a label (Annotations tab) which will help you navigate and query. Call it what you like. E.g. “first kinase of pathway” or “kinase activity by daf-2 during dauer”.

Under Types, put “encoded\_by some <GeneProduct>” (you must have the gene products for the organism loaded as classes). Put “occurs\_in some <CC>”.

Create a new individual of type <BP>, where this is the context. Then go back to your MF instance and under “object property assertions” add a part\_of link.

Create the next MF. Then go back to your first and add a “directly\_activates” link or similar between them.

And so on. Periodically sync the reasoner and check your work in the Lego tab.

Save frequently, and commit to svn (location TBD).

1. <http://code.google.com/p/owltools/downloads/list> [↑](#footnote-ref-1)
2. <http://code.google.com/p/owltools/source/browse/#svn%2Ftrunk%2FLego%2Fexamples> [↑](#footnote-ref-2)