

Parameter Interaction

This page is a follow up to geotools-devel discussion asking the important question: How to represent parameters that are expected to capture an interaction?

Examples:

- Parameter supplying an Attribute, needs to know which Parameter is supplying the schema (so we can show a nice drop-down list)
- Parameters that are required to work together in a group
- Parameters that are exclusive (only one can be used in a group)

Original email:

I would like to propose two simple modifications to the Parameter class, to enhance the semantics of geoprocesses. They both can be implemented as new hints, and made available through the DescribeParameter annotation

- A hint describing dependence of one parameter to another. The most typical use for that would be to indicate the relation between a String parameter representing an attribute and a FeatureCollection. That would allow the user interface to make it easier for the user to select the field, as it could be taken from the list of available ones for the selected FC

- A hint describing the group of optional one a parameter belongs to. Some processes have a set of 2 or 3 parameters, from which just one has to be used (see the ContourProcess for example). If these grouping is not explicitly described, the UI will let the user fill all parameters or none of them, instead of enforcing using only one of them.

As you can see, it is a very simple thing, but I think that would add more robustness to the process semantics. Adapting existing processes to this change is trivial and should take little time to do.

Looking forward to knowing your opinion on this.

Regards

Victor

Here are the code examples.

I am considering combining the two ideas:

a) Use GROUP to "indicate" the parameters in the same group work together

b) Use INTERACTION to indicate how they interact, if not interaction is specified then the parameter considered the subject or topic of the group (and is thus "enabled", or described by "metadata", or whatever)

Here is what that looks like ...

Metadata

DEM Metadata

```
Parameter<File> DEM = new
Parameter<File>("dem", File.class, "DEM
File", "File to import",
    true, 1, 1, new KVP( GROUP, "elevation")
);
Parameter<File> METADATA = new
Parameter<File>("metadata", File.class,
"Metadata", "DEM Metadata",
    false, 0, 1, new KVP( GROUP,
"elevation", INTERACTION, "metadata" ) );
```

Exclusive

Exclusive

```
Parameter<Double> interval= new
Parameter<Double>("interval", Double.class,
"Interval ",
    "Interval between contour values (ignored
if levels parameter is supplied)",
    false, 0, 1, new KVP( GROUP, "contours",
INTERACTION, "exclusive" ) );
Parameter<Double[]> levels= new
Parameter<Double[]>("levels", Double[].class
, "Fixed contour elevations",
    "Values of levels at which to generate
contours",
    false, 0, 1, new KVP( GROUP,
"contours", INTERACTION, "exclusive" ) );
```

Enable

Enable

```
Parameter<Boolean> SNAPPING = new
Parameter<Boolean>("snapping",
Boolean.class, "Enable Snapping",
    "True to enable snapping, requires use of
'distance' parameter",
    true, 1, 1, new KVP( GROUP ="snap",
INTERACTION, "enable" ) );
Parameter<Double> DISTANCE = new
Parameter<Double>("distance", Double.class,
"Snapping Distance",
    "Snapping distance used, required when
'snapping' parameter is True.",
    false, 0, 1, new KVP( GROUP = "snap" ) );
```

Attribute and Expression

This example is interesting as both DISSOLVE and ATTRIBUTE "look up" in their group for the parameter providing a "schema". Attribute allows more than one attribute to be defined, using the same Transform Definition thing as the transform process.

Attribute

```
Parameter<FeatureCollection> FEATURES = new
Parameter<FeatureCollection>("features", FeatureCollection.class, "Features",
    "Features being dissolved",
    true, 1, 1, new KVP( GROUP = "dissolve",
INTERACTION, "schema" ) );
Parameter<String> DISSOLVE = new
Parameter<String>("dissolve", String.class,
"Dissolve Attribute",
    "Attribute used to form groups when
dissolving",
    true, 0, 1, new KVP( GROUP = "dissolve" )
);
Parameter<TransformDefinition> ATTRIBUTE
= new
Parameter<TransformDefinition>("attribute",
TransformDefinition.class, "Attribute",
    "Attribute defined using an aggregate
function (sum, min, max, first, average)",
    false, 0, Integer.MAX_INTEGER, new
KVP( GROUP = "dissolve" ) );
```