

Separate general complex feature classes from gt-app-schema

Contact:	Niels Charlier
Tracker:	GEOT-4344
Tagline:	

- [Description](#)
- [Status](#)
- [Tasks](#)
- [API Changes](#)
 - [Documentation Changes](#)

Children:

Description

The idea is to split the app-schema module in three:

- The first part would contain everything that helps with complex features in general and can be moved to gt-main: mainly builders
- The second part would contain everything that helps with complex features in general and depends on XSD: evaluating filters against them (property accessor, x-path evaluation), building complex feature types from XSD. These classes do not rely on specific schemes like GML, etc... apart from XS.
- The third part will contain anything that is app-schema specific: creating complex feature datastores with *mapping files*.

The third part would continue as the gt-app-schema module. The first part will be moved to gt-main. More specifically, the following classes will be moved:

- org.geotools.feature.TypeBuilder
- org.geotools.util.ComplexAttributeConverterFactory
- org.geotools.feature.AttributeBuilder (PARTLY)
- org.geotools.feature.Types (PARTLY)

The second would split off and become a new module 'gt-complex'. More specifically, this 'gt-complex' module will consist of the following classes that are now in the gt-app-schema module.

- org.geotools.filter.expression.FeatureProperty + complete org.geotools.feature.xpath package
- complete org.geotools.feature.type package
- org.geotools.data.complex.filter.XPath (PARTLY)
- org.geotools.data.complex.FeatureTypeRegistry (PARTLY) + depending classes (for building types from xsd)
- org.geotools.data.complex.config.EmfAppSchemaReader

Considering the 'partly' ported classes.

- AttributeBuilder - will still need an extended version in app-schema to support the NonFeatureTypeProxy which uses mappings
- XPath , Types - are just classes of different static methods that need to be split in two
- FeatureTypeRegistry will be made more generic using a helper interface 'FeatureTypeRegistryHelper', app-schema can have its specific implementation

The gt-complex module will have a dependency on the module gt-app-schema-resolver.

Status

This proposal is currently being voted on:

- [Andrea Aime](#) +1
- [Ben Caradoc-Davies](#) +1
- [Christian Mueller](#)
- [Ian Turton](#) +1
- [Justin Deoliveira](#)
- [Jody Garnett](#)
- [Simone Giannecchini](#)

Tasks

This section is used to make sure your proposal is complete (did you remember documentation?) and has enough paid or volunteer time lined up to be a success

	no progress		done		impeded		lack mandate /funds/time		volunteer needed
--	-------------	--	------	--	---------	--	--------------------------	--	------------------

1. NC: move/copy all the classes (specified above) into new module 'gt-complex'
2. NC: refactor app-schema to use the gt-complex module and the new featuretyperegistry
3. NC: Pull request for [GEOT-4344](#)
 - <https://github.com/geotools/geotools/pull/88> complex model refactoring
4. NC: review the geotools documentation and update
 - upgrade instructions for 9.x will need to mention this new dependency

API Changes

The API will not change because of this.

Documentation Changes

gt-complex should get its own page in the GeoTools User Documentation. The application-schema documentation page says "Key to the configuration of the app-schema module is the *mapping file*". The gt-complex module provides support for complex features but has nothing to do with mapping files.