

Feature Model Proposal

This proposal has been accepted and is the subject of the following active development:

- [Feature Model](#) (API including nice diagrams)
- [Feature Model Branch](#) implementation status of the fm branch

This proposal was completed in the [2.2.x](#) time frame, and is the subject of development in [2.3.x](#)

Proposal

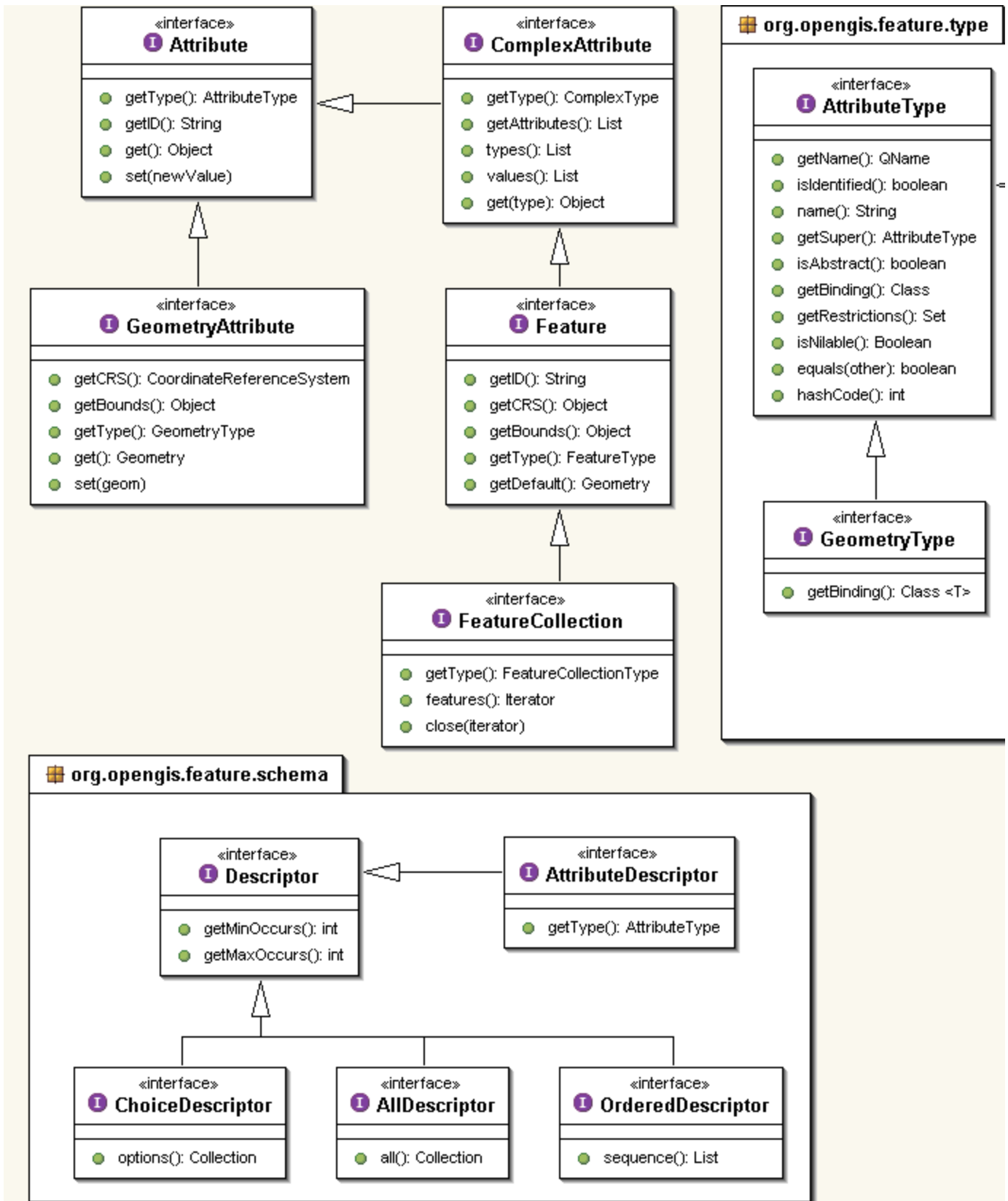
Based on the [Feature Model Discussion](#) we are having a bit of fun making the separation of schema from type.

Notes:

- javadocs are used when needed to explicitly capture what is going on
- Java Bean conventions are used to access simple properties
- Collections conventions are used to traverse contained content, or for convenience methods.

This proposal is based on a split between Content, Type and Schema. The exciting part is - when you are working with "SimpleFeatures" you don't have to look at any of the Descriptor information (it simply does not apply to your problem domain).

- [feature](#)
- [feature/type](#)
- [feature/schema](#)

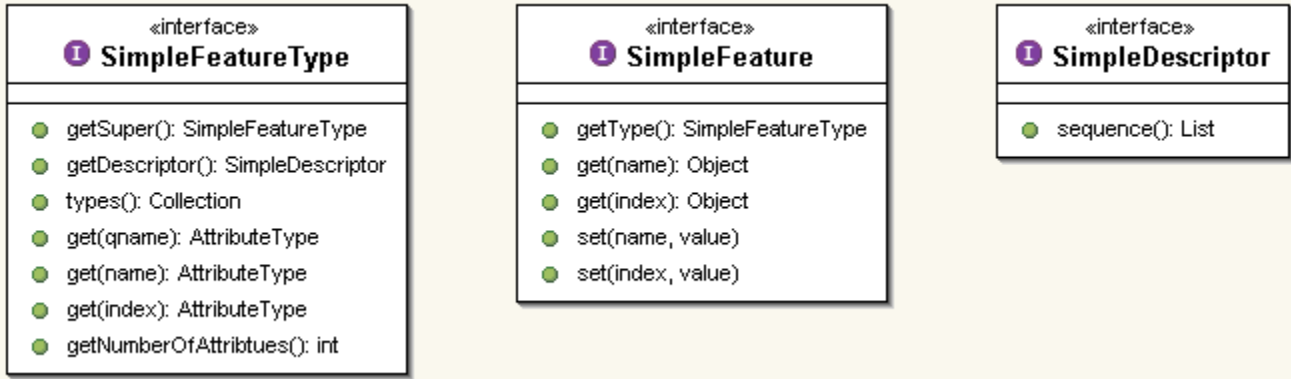


(Javadocs have not been generated at this time)

Goal Simple API for Simple Content

- [feature/simple](#)

I would like to "reward" those with simple content with a simple API. If your data is a shapefile (or otherwise "Flat"), you should be able to ignore all Complex & Multiplicity issues.



The gold standard would be able to work from the Type description, ignore the middle layer and actually have the simple content view of life that XPath maintains:

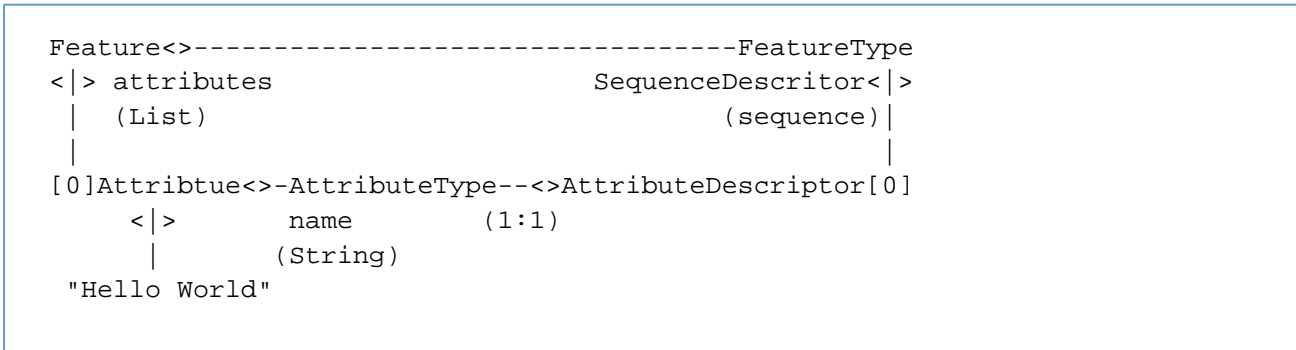
	Java	Simple
Content:	Object	Object, SimpleFeature
Type:	Class	AttributeType, SimpleFeatureType

This is pure extension and does not impact on the modeling "power" of the Feature Model.

Complex Hello World

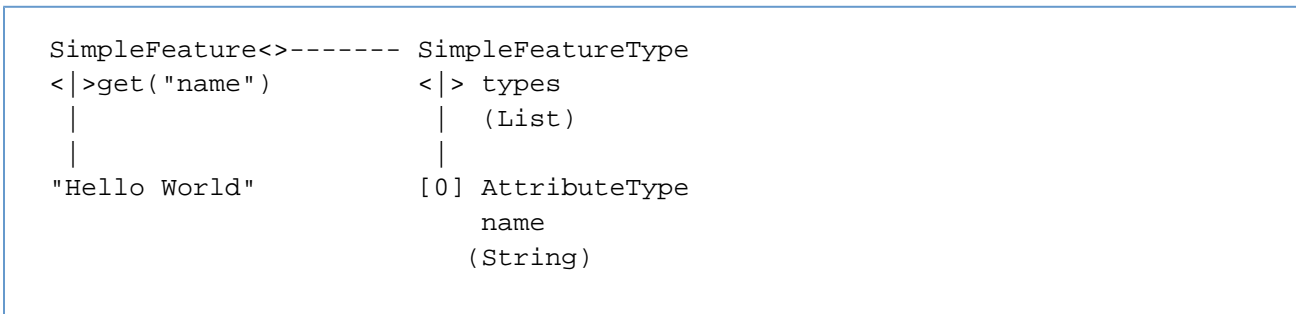
Consider the following information: [name="Hello World"]

When you consider the complete Schema, the picture would look like this.



Simple Hello World

Given the 1 to 1 restriction for SimpleFeature we can just consider the following:



Schedule

There are four main projects that need to be kept in the loop for this proposal to be a success:

Project	Contact	Involvement
GeoTools	Gabriel Roldán	
GeoServer	Chris Holmes	sponsor target (see ComplexDataStore project)
User- friendly Desktop Internet GIS	Jody Garnett	(can start in November)
GeoAPI		Need proposal for November OGC meeting

There are a couple other projects that are closely following this work, to see if the end result will be beneficial.

⚠ This schedule is a Negotiation - I will ask Project Leads to Review ⚠

Notes:

- Gabriel has fixed deadlines
- Intergration with UDIG cannot start until early November
- GeoTools PMC have written a [Feature Model Letter of Support](#)
- GeoAPI has a hard deadline of an OGC meeting in November, work needs to be available mid October

Entries already denoted by:

CD (Complex DataStore) entries are already in Gabriel's plan for GeoServer

GA (GeoAPI) Are needed for GeoAPI acceptance

GT (GeoTools) Are needed for GeoTools acceptance

GS (GeoServer) To be negotiated with GeoServers mailing list?

UD (UDIG) To be negotiated with UDIG mailing list?

Part of the reason we are able to ask the GeoTools/GeoServer/UDIG community to consider this proposal is for the opportunity to harmonize with GeoAPI.

July 22nd - Completed

1 CD Alignment of complex sco branch with trunk

- Update previous GeoServer 1.2 work to GeoServer 1.3 trunk

August 30th, Completed

2 GT Feature Type mappings suite

- reports with real life examples of GML schemas and documents

September 9th, Completed

3 GT - FeatureType survey, description of current, limitations to be fixed, proposed architecture

- [ProposedAttributeTypeAPI](#)
- The architecture proposal is complete, we are struggling with a few design issues right as part of the next step

September 15th

3 GT FeatureType test suite - implementation of proposed architecture with JUNIT tests proving the approach

3.1 GT Interfaces extended from GeoAPI baseline marking changes as required (hard for a complete overhaul)

3.2 GT Implementation based on in memory data

3.2 GT implement tests proving the business driver examples are covered,

3.3 GT implement tests proving the XPath support for SLD documents are covered

(gs:--) GeoServer 1.3.0 released freeing up resources for GeoServer 1.4.x

October 1st

4 GA Integration with GeoAPI

4.1 GA move interface into geoapi pending

4.2 GA Write up of draft change request

4.3 GT branch of trunk for integration

4.5 GT implementation of shapefile

4.6 GT implementation of postgis

4.7 GT ComplexDataStore, a datastore able to query from JDBC to complex (original 9-16th!)

October 15th

5 GA OGC Change Request

5.1 Port of remaining datastores (only WFS should be hard)

November 1st

6 GT merge with gt trunk, milestone release to community

6.1 GS start of geoserver changeover to milestone release/trunk

6.2 UD udig 1.2.0 released as a stable target for development, frees up team for udig 1.3.x ...

6.3 UD start of udig changeover to milestone release/trunk

6.4 GT GML Production enhancements, let geotools write out GML (original 9-29th!)

November 15th

7 GA OGC Meeting, presentation to Working Group

7.1 GS ongoing geoserver changeover, this time with focus GML production

7.2 UD ongoing udig changeover, this time on random access?

December 1st

8.1 GS initial release of geoserver 1.4.0-M0

8.2 UD initial release of udig 1.3.0-M0

Feb 1st

9.1 UD target release for 1.3.0 w/ complex type support

9.2 GS target for release of 1.4.0 w/ complex type support