Geoscientific
Instrumentation
Methods and
Data Systems

Discussions

The GPlates Geological Information Model and Markup Language

X. Qin[1], R. D. Müller[1], J. Cannon[1], T. C. W. Landgrebe[1], C. Heine[1], R. J. Watson[2], and M. Turner[3]

[1]EarthByte Group, School of Geosciences, University of Sydney, Sydney, NSW 2006, Australia
[2]Geodynamics Team, Geological Survey of Norway, P.O. Box 6315, Sluppen, 7491 Trondheim, Norway
[3]Seismological Laboratory, California Institute of Technology, Pasadena, California, USA

Correspondence to: X. Qin (michael.chin@sydney.edu.au)

GID

2, 365–428, 2012

The GPlates Geological Information Model

X. Qin et al.

Title Page

Abstract | Introduction

Conclusions | References

Tables | Figures

◀ | ▶▶

◀ | ▶

Back | Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion

Abstract

Understanding tectonic and geodynamic processes leading to the present-day configuration of the Earth involves studying data and models across a variety of disciplines, from geochemistry, geochronology and geophysics, to plate kinematics and mantle dynamics. All these data represent a 3-dimensional spatial and 1-dimensional temporal framework, a formalism which is not exploited by traditional spatial analysis tools. This is arguably a fundamental limit in both the rigour and sophistication in which datasets can be combined for geological "deep time" analysis, and often confines the extent of data analyses to the present-day configurations of geological objects. The GPlates Geological Information Model (GPGIM) represents a formal specification of geological and geophysical data in a time-varying plate tectonics context, used by the GPlates virtual-globe software. It provides a framework in which relevant types of geological data are attached to a common plate tectonic reference frame, allowing the data to be reconstructed in a time-dependent spatio-temporal plate reference frame. The GPlates Markup Language (GPML), being an extension of the open standard Geography Markup Language (GML), is both the modelling language for the GPGIM and an XML-based data format for the interoperable storage and exchange of data modelled by it. The GPlates software implements the GPGIM allowing researchers to query, visualise, reconstruct and analyse a rich set of geological data including numerical raster data. The GPGIM has recently been extended to support time-dependent georeferenced numerical raster data by wrapping GML primitives into the time-dependent framework of the GPGIM. Coupled with GPlates' ability to reconstruct numerical raster data and import/export from/to a variety of raster file formats, as well as its handling of time-dependent plate boundary topologies, interoperability with geodynamic softwares is established, leading to a new generation of deep-time spatio-temporal data analysis and modelling, including a variety of new functionalities such as 4-D data-mining.

1 Introduction

Representing heterogeneous geospatial information in an Earth history context is a complex issue. At one extreme, if we attempt to express all spatio-temporal information uniformly in terms of a fixed set of attributes, we risk both missing important attributes that are relevant to only some types of information, as well as coercing a user's view to be constrained to those attributes. At the other extreme, if we accept any combination of attributes and attribute types, there is no guidance about which attributes and their properties are expected or required by plate tectonic software; as a result, attributes will be chosen ad-hoc from project to project, hindering attempts to share data between projects. We, therefore, created the GPlates Geological Information Model (GPGIM) to avoid these pitfalls. An information model is an abstract, formal representation of entities in a constrained domain context. It provides formalism to the description of the chosen domain of discourse without constraining how it is mapped to an actual software implementation. The advantage of using an information model is that it can provide sharable, stable, and organised structure of information requirements for the domain context (Lee, 1999).

Traditional plate-tectonics data have consisted of vector geometries and primitive type attributes such as strings, integers and floating-point numbers. These primitive data types are insufficient to represent the next-generation data calculated and processed by GPlates (Boyden et al., 2011), such as attribute values, which vary as functions of geological time, geological time scales, structured edit-history metadata, dynamic topological plate boundaries (Gurnis et al., 2012), and plate tectonic flowlines. The GPGIM has been defined in a plate tectonic context and represents a unified framework in which relevant types of geological data are attached to a common plate tectonic reference frame. It lays a foundation for the development of plate tectonic spatio-temporal data analysis and modelling. The GPGIM specifies the geologic, geophysical and paleo-geographic entities that GPlates simulates, the conceptual building-blocks which GPlates defines, and the processing and simulation constraints to which

The GPGIM is currently able to represent both legacy geological data, as well as various next-generation data for state-of-the-art functionality being developed in GPlates,
for example, the time-dependent geometries defined by geospatial topological networks (Gurnis et al., 2012). The GPML is a GML application schema (Cox et al., 2002),
which implements GPGIM with XML schema in the data serialization context. All the
data processed by GPlates can be saved as GPML files which essentially are XML
files with a GPML schema (Lake, 2004). The GPML schema contains all the structural information about the data contained in the XML files, therefore it is very easy
for other software systems to understand and process these XML files. Furthermore,
GPML is based on GML, a widely accepted standard. This standard-compliant nature
makes GPML even more approachable. And reusing GML elements alleviate the pain
of building various XML parsers to retrieve information from xml files.

2 The GPlates Geological Information Model

2.1 Feature

A key concept in GPGIM is that of the feature: an entity of interest that might be labelled
or referenced. The concept of the feature is borrowed from traditional Geographic Information System (GIS) (Bonham-Carter, 1994), with the important distinction that a
feature in GPGIM is a generalisation of the traditional GIS concept of a feature: while a
GIS feature is an entity of interest which has a geographic location (and usually some
sort of identity or label), in GPGIM, every entity of interest whether spatial or non-spatial
– is modelled as a derivation of feature (Fig. 1). All data and information modelled in
GPGIM take the form of feature instances, or attributes of feature instances.

The feature Id and revision Id in the base class reveal an important concept in
GPGIM-feature reference. To enable persistent inter-instance feature reference, a

universal unique identification of feature is needed. Any feature instance can be identified explicitly by feature ID and revision ID no matter in what form the feature instance is. For example, the feature instance could be an object instance in computer memory, serialized data file on hard disk or just a conceptual entity in a UML diagram, etc (Larman, 2004). The feature ID remains the same during the whole life cycle of a feature. The revision ID changes each time a new revision of the feature has been created. With both feature ID and revision ID, a specific revision of a feature instance can be referenced if desired. The feature reference is a pre-requisite for the feature-oriented data-structures that enable GPGIM to represent and process topological geometries and deforming meshes.

In contrast to some geospatial data formats, in which data are classified primarily by geometry (as "point-data", "line-data", etc.), the type of a feature in GPGIM is independent of the geometries which the feature contains, determined instead by the geological, geophysical or geographic characteristics of the entity which the feature represents.

2.1.1 Reconstruction features

The reconstruction feature is an integral part of plate-tectonic reconstruction process. It represents the model which determines the plate hierarchy and the history of plate-tectonic reconstructions, by means of finite rotations. In this case the geographic location associated with the rotation data is a rotation pole (Greiner, 1999).

In classical plate tectonics, the tectonic plates are assumed to be internally rigid (Müller et al., 2008) and are in constant motion with respect to one another. The relative motion of two plates can be described by finite rotations.

And Euler's rotation theorem states (Euler and Sten, 2012):

"*In whatever way a sphere is turned about its centre, it is always possible to assign a diameter, whose direction in the translated state agrees with that of the initial state.*"

The GPlates Geological Information Model

X. Qin et al.

The GPlates Geological Information Model

X. Qin et al.

The intersection of the axis passing through the centre of the model sphere with the sphere's surface is the Euler pole. A rotation by a given angle around a fixed Euler pole is a finite rotation. A finite rotation that displaces a plate from its present-day position to its reconstructed position at some instant in the past is called a total reconstruction pole (Cox and Hart, 1986).

The TotalReconstructionSequence (Fig. 2), as a generalization of Reconstruction-Feature, contains a sequence of total reconstruction poles, which have been defined to describe the plate's motion through time by sampling its relative displacement at key instants in the past. Each sampling is represented by a TimeSample < FiniteRotation > instance (Fig. 2). The total reconstruction poles for any given time can be calculated by interpolation. Cox and Hart have described the mathematical rules in Cox and Hart (1986). Using finite rotations is only one of the possible approaches to model plates-tectonic reconstructions. GPGIM is an open model; new derivations of ReconstructionFeature can be defined to introduce new reconstruction methods as needed.

2.1.2 Reconstructable feature

The most common features in GPGIM are reconstructable features. The reconstructable features can be reconstructed to different paleo-geographic positions at different geological times, according to the plate-motion model. A reconstructable feature possesses special properties which enable it to be associated with the motion of a tectonic plate – either a reconstruction plate ID or a reconstruction method for more complex forms of reconstruction. The geometries in a ReconstructableFeature are stored in present-day coordinates. The reconstructed geometries are calculated at runtime when they are needed. At present, more than 50 types of ReconstructableFeature are defined in GPGIM. There are four main subcategories of ReconstructableFeature (Fig. 3).

Tangible feature

The tangible feature is a type of feature that abstracts geographic entities. For example, the seamount feature, which is derived from TangibleFeature directly, represents a volcanic cone rising from the ocean seafloor that does not reach to the sea surface (Wessel, 2009). The GPGIM contains a comprehensive collection of tangible features (Fig. 4). Each feature has been modelled carefully by specifying a unique property set, which represents the geographical characteristics of the feature.

Thus, there are as many different types of features in the GPGIM as types of entities represented, ensuring that the feature which models an entity has exactly the properties it needs, no more and no less. The naming of the feature types follows geological nomenclature, to maximise the likelihood that a geologist will understand what a feature represents, when presented with the feature type. For example, an ocean floor isochron is represented by the Isochron feature; a subduction zone by the Subduction Zone feature; and a dynamic topography field by the Dynamic Topography feature.

Artificial features

In contrast to TangibleFeature, the ArtificialFeature type deals with geometries that have been arbitrarily created, or possibly derived from a TangibleFeature which no longer exists. For example, InferredPaleoBoundary is a plate boundary that is assumed to have existed, even though it has long gone since it has been subducted at a deep sea trench and recycled into the convecting mantle.

There are four types of feature in this category (Fig. 5):

Topological boundary guide

Different from a TopologicalClosedPlateBoundary which is defined exclusively via references to ReconstructableFeatures' geometry, the TopologicalBoundaryGuide feature

type contains user-defined geometry (lines or points) which defines the sections of a TopologicalPolygon.

Unclassified feature

Sometimes, data from other research domains or communities need to be imported into GPGIM to collaborate. This UnclassifiedFeature type allows alien data to be handled in GPGIM decently and therefore makes GPGIM more powerful and flexible.

2.1.3 Topological feature

A topological feature is a feature which lacks a reconstructable geometry of its own. A topological geometry can be calculated after each reconstruction by linking together the reconstructed geometries of other referenced feature instances. Topological features are some of the most exciting next-generation data in GPGIM. The most prominent topological feature in GPGIM at present is the Topological Closed Plate Boundary, which as its name suggests is a feature whose topological geometry represents a closed plate boundary. This closed plate boundary is a time-dependent shape-changing polygon (also known as a Continuously-Closing Polygon – CCP) that encloses a region corresponding in some practical sense to a tectonic plate (Gurnis et al., 2012). In the future, topological features will also be used to model plate deformations in GPGIM.

2.1.4 Instantaneous features

In contrast to the dynamic reconstructable features are *instantaneous features*, static snapshots of entities at an instant in geological time. These features correspond most closely to the static geospatial features of traditional GIS software. Instantaneous features are intended primarily as an interchange format that enables reconstructed features to be exported to, or imported from, other geospatial information model, which lacks the ability to process plate-tectonic reconstructions or time-variance. An

Title Page

Abstract | Introduction

Conclusions | References

Tables | Figures

◄◄ | ►►

◄ | ►

Back | Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion

The GPlates Geological Information Model

X. Qin et al.

instantaneous feature preserves the reconstructed geometries and the current values of any time-dependent property values in a form that does not require any special processing by the recipient information model. A time-sequence of instantaneous features can be integrated into a single reconstructable feature with a continuous history. Every type of reconstructable feature in the GPGIM has an instantaneous analogue.

2.2 Feature property

Another important concept in the GPGIM is the feature property. Each concrete feature type has been associated with a group of unique feature properties, which means that each feature instance must contain and only contain a set of predefined properties, unless the property is optional. By enforcing this, GPGIM can guarantee that the data loaded into the model is valid. Each feature property contains a property name and a time dependent value (Fig. 6).

2.2.1 Property name

Instead of allowing arbitrary strings as property names, GPGIM defines a finite set of property names. Only these predefined names are considered as valid. And the property names carry more information than just identification. For example, the name reconstructionPlateId indicates the data contained in this property is a plate id, which can be used in plate-tectonic reconstruction.

The property name implies a real property type, which in turn affects the behaviours of the feature containing this property. For instance, to determine if a feature is reconstructable, we search for either of the two properties reconstructionPlateId and reconstructionMethod. Any feature instance that has either of these properties can be assumed to be reconstructable.

2.2.2   Property value

Just as features have types, which determine the properties contained within a particular feature instance, so, too, do property-values have types, which determine the range of values that may be contained within a given property. Being a GML application schema (Lake, 2004), the GPGIM incorporates the property-value types defined by GML, which in turn incorporates the XML primitive data types.

The XML primitive data types include, but not limited to, xs:string, xs:boolean, xs:integer and xs:double. For example, a property-value of type xs:integer contains only a single integer value, while a property-value of type xs:string contains a string of text. On top of these primitive data types, GML defines additional property-value types as building-blocks for GIS data, such as geometries (gml:Point, gml:MultiPoint, gml:LineString and gml:Polygon) and temporal primitives (gml:TimeInstant and gml:TimePeriod) (Bray et al., 1997; Cox et al., 2002).

Finally, the GPGIM defines a number of GPlates-specific property-value types. We have classified these GPlates-specific types into four main categories, namely constant value, irregular sampling, piecewise aggregation and math function of time (Fig. 7).

– Time-dependent property values
It is a fundamental assumption built into the GPGIM that the measurable characteristics of geological entities can change over geological time. In a feature-oriented model, this is represented as variation in the feature property values over geological time. The TimeDependentPropertyValue is defined as the root of inheritance hierarchy to represent the concept of this assumption. All property values in GPGIM are derivations of time-dependent value.

– Constant value
The constant value type is the simplest form of variation of a time-dependent value. It represents a type of value which remains constant through geological time.

Title Page

Abstract   Introduction

Conclusions   References

Tables   Figures

◄◄   ►►

◄   ►

Back   Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion

- Irregular sampling
  The irregular sampling represents a sequence of time-samples, each of which was sampled at some geological time-instant. An irregular sampling may also possess an optional interpolation function, to interpolate between the values of adjacent time-samples.

- Piecewise aggregation
  The piecewise aggregation provides a way to compose different property value types to maximize the flexibility of GPGIM. The piecewise aggregation property value contains a sequence of zero or more time-windows, each of which is defined for a particular geological time-interval and contains an instance of time-dependent property value type.

- MathFunctionOfTime
  The MathFunctionOfTime represents a variation algorithm that may be expressed mathematically as a function of geological time. The property value is calculated at a given time at runtime.

3 The GPlates Markup Language (GPML)

With the rapid development of internet, collaboration and resource sharing are playing a more and more important part in the success of academic research. How to exchange and share scientific data efficiently is arising as a key issue. To alleviate the disorder brought by heterogeneous data, GML was conceived (Lake, 2004). GML is a modelling language for geographic systems, as well as an encoding specification for the serialisation of GML data to XML documents. GML is intended to provide a standardised, interoperable base for defining application-specific geospatial languages. The GML specification is defined and maintained by the Open Geospatial Consortium (OGC®). GML was adopted as an ISO standard in 2007 (Lake, 2004). GML defines information model building-blocks including: geometries; temporal primitives (time-instants and

Full Screen / Esc

Printer-friendly Version

Interactive Discussion

time-periods); units of measure; and coordinate reference systems (Lake, 2005). These building-blocks ease the process of interoperability both by providing recognisable, well-defined data components and by standardising the XML representations of these components, to simplify the parsing of foreign data.

The GPML is a GML application schema which is designed to describe and exchange plate-tectonic data within a community of interest. It enables collaboration, data sharing and grid computing in plate-tectonic domain. GPML is an extension to GML; it adopts the feature-oriented modelling approach and builds upon the GML primitives to maximise interoperability with other GML based application schemas such as GeoSciML (Sen and Duffy, 2005). Another significant advantage of being based on GML is that the data described by GPML can be transmitted across Web Feature Service (WFS) networks by default(Peng and Zhang, 2004). It makes the data sharing and transmission easy and fuels a wide range of collaboration and grid computing. The WFS specification (Panagiotis, 2005) defines a web service interface to query, transmit and manipulate geospatial data across the Internet or other networks. GML is the default encoding for the transmission of geospatial data to and from WFS servers.

3.1 Feature and feature collection

Every concrete GPML feature is derived from GPML AbstractFeature (Fig. 8) which, in turn, derives from GML AbstractFeatureType. The substitutionGroup attribute in xml element AbstractFeature indicates that all GPML abstract features are substitutable for the GML AbstractFeature, which is defined in GML specification. The inheritance hierarchy and the substitutionGroup attribute are essential to serving GPML data through WFS networks.

The web feature service provides data by responding to GetFeature requests from client. The root element of the response must be a wfs:FeatureCollection (Fig. 9) (Panagiotis, 2005). The wfs:FeatureCollection derives from gml:AbstractFeatureCollection.

The GPlates Geological Information Model

X. Qin et al.

Full Screen / Esc

Printer-friendly Version

Interactive Discussion

From the definition of gml: AbstractFeatureCollection (Fig. 10) and gml:featureMember (Fig. 11) we can conclude that, essentially, the abstract feature collection of gml, as the name suggests, is a collection of gml:AbstractFeature (Cox et al., 2002).

All GPML features are derived from GML AbstractFeature and, therefore, any GPML feature instance is a valid featureMember in a FeatureCollection which is prepared by WFS server and sent back to client as a response to a previous GetFeature request (Panagiotis, 2005).

Figure 12 shows an example of concrete GPML feature definition. The IsochronType is derived from TangibleFeatureType, which, in turn, is derived from gpml:AbstractFeature. The conjugates in IsochronType are feature references which link to the "pair"s or "twin"s of this isochron feature. The conjugatePlateIds are plate Ids, either general-purpose plate Id or specific plate Id for the motion of a cluster of features, that has the "pair"s or "twin"s of this isochron. The unclassifiedGeometry is an optional geometry whose purpose is not given or unclear. The centerLineOf actually defines the geometry of this isochron.

More than 100 concrete feature types have been defined in GMPL. A full definition of all GPML feature types can be found at this URL – http://www.earthbyte.org/Resources/GPGIM/public/gpml.xsd.

3.2 Property value types

The property value types are defined as a direct mapping from GPGIM into xml schema and all of them are derived from gpml:TimeDependentPropertyValueType. The xml schema definitions for Constant (Fig. 13), Irregular Sampling (Fig. 14), and Piecewise Aggregation (Fig. 15), value types are straightforward. The fourth value type, Math-FunctionOfTime, has yet to be implemented.

3.3 Geometry

The geometries in GPML reference GML geometries directly. By using building blocks from GML, GPML geometries can be parsed and processed easily by other GML applications. In GPML, geometries are not abstract geometrical shapes anymore; they are a part of the features' characteristics.

3.3.1 The centerLineOf geometry

The centerLineof geometry represents the centreline of a geological feature. It is generally using gml:LineString, with the possibility of gml:MultiGeometry or gml:MultiCurve (Fig. 16).

3.3.2 The outlineOf geometry

The outlineOf geometry specifies the closed or partial outline of a geological feature. The possible geometry types are gml:MultiGeometry, gml:LineString and gml:Polygon (Fig. 17).

3.3.3 The errorBounds geometry

The errorBounds geometry defines the error boundary of a geological feature. It can be anything substitutable for gml:AbstractGeometry (Fig. 18).

3.3.4 The boundary geometry

The boundary geometry represents the polygon boundary of a geological feature (Fig. 19).

3.3.5 The position

The position is a point in latitude and longitude indicating the physical location of a geological feature (Fig. 20.).

3.3.6 The unclassifiedGeometry

The unclassifiedGeometry is a geometry which has been defined for the feature where the purpose is not given or unclear. The unclassified geometry type can be used to represent the geometries imported from other domains of discourse (Fig. 21).

4 Implementation in GPlates

GPlates is an open source software application, which is developed by an international collaboration led by the Earthbyte group at the University of Sydney, and also including the California Institute of Technology and Geological Survey of Norway. It implements GPGIM and uses GPML to serialize and exchange data. Being well-equipped with GPGIM and GPML makes GPlates a state-of-the-art scientific research and teaching assistance tool with outstanding capabilities of calculation and interactive visualisation and manipulation of plate-tectonic reconstructions through geological time and deep-time data mining/analysis (Williams et al., 2012).

4.1 Reconstruction tree

In GPGIM, the ReconstructionFeature represents the concept of an entity which contains all information pertaining to the reconstruction process. In the real world, the concept is embodied as a list of finite rotations in a rotation file. Figure 22 shows a snippet of rotation file. Each line in rotation file represents a finite rotation of moving plate relative to the fixed plate. The first column specifies the moving plate id. The second column is geological time in Million Years (Ma). The following three columns define

the latitude and longitude of Euler pole and rotation angle. The sixth column is fixed plate id which is followed by comments.

When a rotation file is loaded in GPlates, a set of instances of TotalReconstruction-Sequenece, which is a derivation of ReconstructionFeature, are created. And they will be used to calculate reconstruction tree at each given geological time.

Fiture 23 shows a small subset of reconstruction tree created from "Global EarthByte GPlates Rotation" file. The following steps show how to calculate the total reconstruction poles of a given plate by using the reconstruction tree at a given time:

1. Locate the plate id in the reconstruction tree at given time.

2. Calculate the rotation of moving plate relative to its parent (fixed plate).

3. Move upwards along the tree to calculate the total reconstruction poles of its parent.

4. Repeat the step 2 and 3 until we reach the root of reconstruction tree.

5. Compose the rotation relative to its parent and its parent's total reconstruction poles.

Figure 24 shows a screenshot of GPlates reconstruction tree dialog which shows the finite rotations calculated from "Global EarthByte GPlates Rotation" file.

4.2 Reconstruction

Once the reconstruction tree has been built, the reconstruction process is very straightforward. The component diagram (Fig. 25) illustrates the process.

1. Reconstruction engine gets reconstructable features from feature store

2. Reconstruction engine acquires the reconstruction tree at given time from reconstruction features.

The GPlates Geological Information Model

X. Qin et al.

3. Reconstruction engine rotates the geometries of each reconstructable feature according the total reconstruction poles calculated from the reconstruction tree base on the feature's plate id.

4. Reconstruction engine returns the reconstructed features at given time to client.

4.3 Non-standard reconstruction methods

Most reconstructable features in GPlates are reconstructed by applying a rotation corresponding to the total reconstruction pole for that feature's reconstructionPlateId property (Sect. 2.1.2). GPlates also allows more complex reconstructions to be performed where for example the present day geometry is reconstructed according to some more complex function of several plate ids, times, or other user-provided information.

An example of such a feature is the gpml:flowline, which is used to represent a synthetic flowline. Synthetic flowlines represent the movement of material away from a spreading centre and can be used to test reconstruction models by comparing with fracture zones.

Flowlines are generated using the half stage rotations between adjacent plates at a series of user-provided reconstruction times. To create a flowline feature in GPlates the user provides a seed point representing a location at a spreading centre. Rather than providing a single reconstructionPlateId, the user provides the plate ids of the plates on either side of the spreading centre, and a series of times; a series of half stage rotations are generated for adjacent times in this series.

At any given reconstruction time, the flowline is generated from the seed point by applying the series of half stage rotations to produce a series of points. These points (nodes) represent the flowline and are visualized on the GPlates globe as a polyline connecting the flowline nodes.

4.4 Dynamically typed feature

In GPGIM, more than 100 feature types have been modelled. Having a large number of feature types makes GPGIM very comprehensive in terms of describing real world geological entities, but on the other side it imposes a huge effort on software development.

Instead of making each feature type a C++ class derived from a proper base class in the class hierarchy, a universal feature handle type has been introduced to handle all feature types and everything inside a feature has been defined as a feature property. It is the properties contained in a feature which determine the feature type, instead of the C++ class type. This approach essentially reflects a form of dynamic typing, although C++ is a static typing language in nature.

The approach that the properties determine feature types is called "duck typing", a style of dynamic typing, following James Whitcomb Riley who said (Heim, 2007):

> "*When I see a bird that walks like a duck and swims like a duck and quacks like a duck, I call that bird a duck.*"

For example, rather than defining a ReconstructableFeature base class in C++ and deriving all concrete reconstructable feature types from this base class (which would result in a large number of C++ classes since there are more than 50 reconstructable features defined in GPGIM), GPlates instead uses a universal FeatureHandle type and searches for two properties, reconstructionPlateId and reconstructionMethod, in the FeatureHandle object. Any feature instance that has either of these properties is deemed by GPlates as a reconstructable feature.

By using "duck typing", the development cycle of GPlates has been shortened significantly and it makes GPlates more flexible when expanding GPGIM. The "duck typing" even makes it possible to change feature type over geological time by repopulating the feature's properties.

However, there is one weak point in "duck typing". To some extent, the "duck typing" is error-prone. With a strongly and statically typed approach, the compiler can help developers with type-checking; hence it's much harder to use an unexpected object

type (Abadi et al., 1995). But, with "duck typing", it is the developers' responsibility to make sure the type correctness. It in turn demands GPlates developers to have a much wider understanding of source code in order to avoid type misuse. To alleviate the pain, a FeatureType object has been added into FeatureHandle class, so that in some circumstance, when the exact feature type is crucial, we can be assertive.

4.5 Plate-motion velocity and geographic coverage

Another interesting implementation in GPlates is mesh-node points and plate-motion velocity. The global mesh-node points distribute evenly on the surface of globe and all tectonic plates are floating above the imaginary mesh. The mesh-node points act as "tracers" for the plate motion. At each reconstruction time, GPlates can calculate an instantaneous angular velocity for each mesh-node point. The velocities of mesh-node points inside a tectonic plate can illustrate the plate-motion in an explicit way. Such a plate-motion velocity field may be exported from GPlates to be used as a kinematic boundary-condition to a geodynamic mantle-convection model such as CitcomS (Tan et al., 2007). This enables GPlates to link plate kinematics to geodynamic models.

The combination of mesh-node points and velocity data essentially is a coverage. A coverage, as defined by OGC and ISO/TC 211, is a function describing the distribution of some set of properties over a spatial-temporal region. The three key aspects of a coverage are the domain set (a group of geometry or temporal objects), the range set (an array of arbitrary value objects), and a coverage function (a mapping of every object in the domain set to a value in the range set) (Lake, 2004). In our velocity case, the mesh-node points are the domain set of a plate-motion velocity field. The velocity data associated with mesh-node points can be deemed as the range set. And the mapping between mesh-node points and velocity is the coverage function.

Full Screen / Esc

Printer-friendly Version

Interactive Discussion

4.6 Rasters

GPlates supports the reconstruction and visualisation of geo-referenced raster image data. Also supported is raster data analysis (in the form of co-registration with point/polyline/polygon seed geometries). Both these applications of raster data are accelerated by commonly available desktop graphics hardware. A present-day raster consists of a single raster image typically representing geospatial data observations on the Earth's present-day surface. GPlates can also create time-dependent rasters by importing a time sequence of raster image files and assigning geological ages to each image. A raster (present-day or time-dependent) is reconstructed by attaching it to a tectonic polygon dataset – raster pixels inside the polygons then rotate with the polygons. This differs from reconstruction of regular geometries (points/polylines/polygons) whereby geometry partitioning, and plate ID assignment, occurs as a once-only preprocess. This enables a raster to remain as a single, un-partitioned GPML feature and to be reconstructable without requiring a reconstruction plate ID property (and is why Raster, in the schema fragment in Fig. 26, inherits from TimeVariantFeatureType instead of ReconstructableFeatureType).

In the geospatial domain many raster image formats (such as netCDF or GeoTIFF) contain not only multi-band raw image data but also information that can be used to correctly position and project a raster onto the Earth's surface, as well as containing general raster metadata. GPlates uses the Geospatial Data Abstraction Library (GDAL), an open-source library for raster geospatial data formats, both for accessing raster data and for raster coordinate transformations. GPlates extracts the raster data into two separate file types. The raw image data is cached into a GPlates-specific multi-resolution, tiled file format for fast streaming access during raster visualisation (one cache file per raster band). The remaining raster image file information is mirrored into a raster GPML file enabling GPlates users to query a raster's coordinate reference system, geo-referencing and general raster metadata through the same query interface used for all existing GPlates feature types. The creation of the raster GPML file occurs

during the raster import process whereby a link is established (and raster information imported) from the raster image file to the raster GPML file. For non-geospatial raster formats (that lack geo-referencing) an affine transform can be manually entered by the user during the import process (with the default coverage being global).

Each raster GPML file contains a single GPML Raster feature that uses a combination of GML and GPML properties and objects to encapsulate a time-dependent geospatial raster. A Raster feature is conceptually a GML coverage feature which is essentially a regular feature with gml:domainSet and gml:rangeSet properties that specify the positions and values of the coverage data. However the schema content model, of Raster (see Fig. 26), derives indirectly from AbstractFeatureType instead of gml:AbstractCoverageType - it cannot derive from both since multiple inheritance is not supported in GML. There are two reasons for doing this. The first is that all GPML features need to inherit properties found in AbstractFeatureType such as featureId. The second is GPlates supports time-dependent rasters and therefore the rangeSet property needs to support a time sequence of image files (which requires the GPML time-dependent property extension). However, in the future, the GPlates reconstruction engine could be used to serve reconstructed raster data via a Web Coverage Service (WCS). In this case Raster will likely be required to derive indirectly from gml:AbstractCoverageType and we would then create a new AbstractCoverageType object that, in turn, derives from gml:AbstractCoverageType by restriction (to redefine the domainSet and rangeSet properties) and also add the necessary properties expected of a GPML feature such as featureId. However this is more of a schema change than an implementation change for GPlates.

The Raster feature defines the time-dependent equivalents of gml:domainSet and gml:rangeSet (replacing the gml prefix with gpml) such that rangeSet is either a ConstantValue or a PiecewiseAggregation of gml:File objects. A similar pattern applies to domainSet with AbstractGeoReferenceType, although currently domainSet is always a ConstantValue which means the same geo-referencing is used for all images in the time sequence (this might change in the future). AbstractGeoReferenceType is an abstract

Full Screen / Esc

Printer-friendly Version

Interactive Discussion

The GPlates Geological Information Model

X. Qin et al.

Full Screen / Esc

Printer-friendly Version

Interactive Discussion

geo-referencing type with the derivations AffineGeoReferenceType and GCPGeoReferenceType (see Fig. 27) depending on whether the raster image file (referenced by gml:File) stores geo-referencing as an affine transform or ground control points.

Each band in the raster is assigned a band name by the user during import into GPlates (stored in the bandNames property of the Raster feature). Each raster has a group of metadata domains for the raster as a whole and for each raster band (for metadata that is specific to a particular band) – see Fig. 28.

Each metadata domain has a dictionary of name/value pairs containing the actual metadata. Each domain has a name except the default domain (see Fig. 29). For example, the NetCDF raster file format supports the SUBDATASETS domain which lists the multi-dimensional data arrays (known as NetCDF variables) contained within.

Typically the coordinate reference system is specified using the 'srsName' attribute of a GML geometry object. This references a publicly available dictionary of spatial reference systems. The same usually applies to implicit geo-referencing geometries (substitutable as AbstractGeoReferenceType) enabling the coordinate reference system of geo-referenced coordinates to be specified. However, in the case of the Raster feature, GPlates obtains the coordinate reference system directly from the linked raster image file (many geospatial raster formats support this). The coordinate reference system is, however, still mirrored in the Raster feature so that GPlates users can query it. This is specified as the coordinateReferenceSystem property whose type is any GML type derived from gml:AbstractCoordinateReferenceSystem (see Fig. 26). GPlates uses the GDAL library to convert the coordinate reference system from Well Known Text (WKT) to GML. Since this property is only queried by the user, a discrepancy in conversion from WKT to GML (due to incomplete GML support) will not result in an incorrectly rendered raster in GPlates.

4.7 Spatio-temporal data analysis

The structured modelling of spatial data and associated metadata varying through time as provided by GPML and the GPGIM provides the underlying foundation necessary to

develop formal data associations and statistical analysis/data-mining tools. This makes it possible to cope with the substantial complexity of simultaneously analysing several datasets in which both spatial geometries and metadata properties vary through time. The data-modelling provides an abstraction to underlying properties and processes, resulting in simple, intuitive and extensible interfaces.

Two components have been developed to allow for the extraction of spatio-temporal associations, namely a coregistration tool and data-mining infrastructure as presented in Landgrebe and Muller (2011). The coregistration tool allows associations between spatial datasets to be defined, abstracted from the feature types involved. Associations include spatial relationships in which distances between features can vary across time, and direct property coregistrations between datasets to assess how a particular property varies as a function of time. The coregistration tool allows users to recursively define a set of associations, resulting in time-series that encode specific dynamics of interest. A data-mining infrastructure has been interfaced with GPlates, using the open-source python-based Orange tool (Demsar et al., 2004). Orange provides a visual user interface that allows users to define a specific data-mining workflow using a library of predefined components. Importantly a plugin facility has allowed components specific to GPlates to be developed, thus providing all the necessary infrastructure to interface the GIS environment with powerful quantitative data analysis.

4.8 Interoperability

According to IEEE Glossary, interoperability is the ability of two or more systems or components to exchange information and to use the information that has been exchanged (Geraci et al., 1991). As an open and flexible software application, GPlates is interoperable with the most widely used GIS systems. Currently, GPlates is using property mapping technique to achieve interoperability between GPGIM and other information models. In the future, Python extensions will be introduced to handle data exchange between GPlates and varieties of systems.

GPlates utilises the GDAL geospatial data abstraction library and is hence capable of reading and writing a number different vector file formats. We will address the interoperability between GPlates and two of them, ESRI Shapefiles and the Generic Mapping Tools (GMT) in the following two sections.

4.8.1 Rotation files

Rotation files are, together with geospatial data, one of the two core data sets which turn GPlates from a geospatial information system into a plate tectonic modelling tool. In order to model plate motions over the geological time in GPlates, the user needs to load a rotation file (*.rot) in order to reconstruct plates to their past positions. Devised in the late 1980's by the PLATES plate tectonic mapping project at the Institute of Geophysics at the University of Texas (Austin), these structured ASCII plain-text files describe a rotation hierarchy of the Earth's tectonic plates as a set of relative Euler (rotation) poles (compare section "Reconstruction Tree" for details). The "PLATES" rotation file format requires only a set of 6 parameters (Moving plate ID, stage pole age, stage pole latitude, stage pole longitude, opening angle, and reference plate ID) and allows for a free form comment (Fig. 22). Since the inception of the PLATES project, this file format became the de-facto standard used by nearly all available plate tectonic modelling software packages (such as the commercially available PaleoGIS, http://www.paleogis.com/DotNetNuke/).

While such a simple structured text format was appropriate at the time and with a very small user base, demands for a more structured and verbose approach have since significantly increased. One of the major drawbacks of the legacy PLATES format is the unstructured comment field and the associated loss of important metadata. These data are needed to properly quantify and verify rotation poles sequences in global plate models through the inclusion of reference data for published rotation poles, statistical information about the accuracy of any given stage pole (Hellinger, 1981), the geological time scale used to relate geological time to absolute time and other useful data such as modification authors.

Ensuring backward-compatibility with legacy PLATES applications, we have invented a new, ASCII plain text-based rotation file format (*.grot) which builds upon the existing PLATES format to describe rotation trees but contains a new set of structured attribute:value tags for consistent and persistent handling of metadata and automated processing of these. The new rotation file format follows ideas of the GMT OGR format (http://www.gdal.org/ogr/drv_gmt.html and http://www.soest.hawaii.edu/gmt5/gmt/pdf/GMT_Docs.pdf), Dublin Core Metadata Initiative and the lightweight MultiMarkDown language (https://en.wikipedia.org/wiki/MultiMarkdown). A detailed specification of the new rotation file format is presented in Supplement (1).

4.8.2   Shapefile

The ESRI shapefile stores nontopological geometry and attribute information for the spatial features in a data set. Since its specifications have been released by ESRI, it has been widely accepted as a de-facto standard for geospatial vector data format and hence provides the interoperability among ESRI and other geospatial information systems. A shapefile consists of "a main file, an index file, and a dBASE® table" (ESRI, 1998). The geospatial information is stored in the main file (*.shp) whereas feature attribute data area stored in the associated dBASE® table (*.dbf), utilising the database management system (DBMS) for microcomputers (Bordwell, 1984). A one-to-one relationship is implicit between the attribute record and the associated shape record; however, the shapefile format does not have a formal information model associated with it. Shapefile attributes can be defined arbitrarily. As only recognized attributes can be accepted and processed by the GPGIM, we designed a property mapping mechanism to import shapefile attributes into GPGIM.

When loading a shapefile into GPlates, the user is presented with a wizard dialog which allows user to specify how shapefile attributes will be mapped into GPlates model (Fig. 30).

When the mapping configuration has been specified by the user, an XML file (Fig. 31) is created to keep the setting in persistent storage so that these attributes can be mapped automatically next time.

Using an XML file to maintain a mapping configuration makes the shapefile not self-contained. Furthermore, there is no specification or standard to tell shapefile users in a plate tectonic context which shapefile attributes are crucial and which ones are optional. Therefore a standard naming convention has been developed which regulates shapefile attributes for use with plate tectonic applications. The standard will make it easier to exchange data in shapefile format between plate tectonic software applications and define attribute names which map directly to the GPGIM (Fig. 32) and is outlined in detail in Supplement (2).

4.8.3   Generic Mapping Tools (GMT) interoperability

The open-source Generic Mapping Tools, available from http://gmt.soest.hawaii.edu, (Wessel and Smith, 1998) are a set of command line programs to process (geospatial) data and generate high-quality Postscript output for publications. Since its inception in late 1980s it has become the de-facto standard tool in large parts of the global geophysics community when processing and disseminating data. GMT routines allow user to read and write ASCII based text files and binary data, such as netCDF.

Through the GDAL library, GPlates is capable of reading and writing GMT-standard xy(z) data files of various different feature collections. Through the GDAL library, GPlates can also write GMT-OGR standard files which encapsulate attribute data consistently and allow interoperability between different formats through the OGR2OGR tool (http://www.gdal.org/ogr2ogr).

4.8.4   GeoSciML interoperability

GeoSciML (Simons et al., 2006) is also a GML Application Schema. It is designed to transfer geologic data, with an emphasis on geologic maps. Since GeoSciML is

associated with an information model, the structure of GeoSciML data is predefined. We use a group of XQuerys (Boag et al., 2003) to retrieve data of interest from GeoSciML data and import it into GPGIM (Fig. 33).

5 Conclusions

The GPGIM along with its implementation, GPML and GPlates software application, represents a number of fundamental differences as compared to previous plate-tectonic reconstruction information models and software applications:

– The use of a rigorous methodology to model geological, geophysical and paleo-geographic features in GPGIM and the GPML application schema, enabling GPlates to serve as a component in a larger GML-based infrastructure.

– Sophisticated information model building-blocks (such as time-dependent property values and inter-feature references) enable expression of advanced and abstract concepts.

– The paradigm-shift of calculation templates for automated post-processing of reconstructions enables new data to be computed "on the fly" based upon plate motions.

Future GPGIM and GPML extensions will include:

– Handling plate deformation

– Improve the interoperability with GeoSciML documents

– Reconstruct GeoSciML features alongside GPGIM features

– Develop the ability to read and write GML-based data, participating in a data grid as a WFS client

With a flexible, expressive, interoperable information model, GPlates and the GPGIM represents the next generation of plate-tectonic information modelling and software – a generation in which plate-tectonic data and applications are an integrated visualisation and processing component within a data grid and computational grid; and a plate-tectonic reconstruction is no longer an isolated result, but a single stage in an adaptable workflow.

Supplementary material related to this article is available online at: http://www.geosci-instrum-method-data-syst-discuss.net/2/365/2012/ gid-2-365-2012-supplement.zip.

References

Abadi, M., Cardelli, L., Pierce, B. C., and Rémy, D.: Dynamic typing in polymorphic languages, J. Funct. Program., 5, 111–130, 1995.

Boag, S., Chamberlin, D., Fernández, M. F., Florescu, D., Robie, J., Siméon, J., and Stefanescu, M.: XQuery 1.0: An XML query language, W3C working draft, 12, available from: http://www. w3.org/TR/xquery/ (last access: 29 June 2012), 2003.

Bonham-Carter, G.: Geographic information systems for geoscientists: modelling with GIS, Pergamon Pr, 1994.

Bordwell, S.: dBASE II – Library use of a microcomputer database management system, Program: electronic library and information systems, 18, 157–165, doi:10.1108/eb046876, 1984.

Boyden, J. A., Müller, R. D., Gurnis, M., Torsvik, T. H., Clark, J. A., Turner, M., Ivey-Law, H., Watson, R. J., and Cannon, J. S.: Next-generation plate-tectonic reconstructions using GPlates, in: Geoinformatics: Cyberinfrastructure for the Solid Earth Sciences, edited by: Keller, G. R. and Baru, C., Cambridge University Press, 2011.

5 Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E., and Yergeau, F.: Extensible markup language (XML), World Wide Web J., 2, 27–66, 1997.

Cox, A. and Hart, B. R.: Plate Tectonics : How It Works, Blackwell Science Inc., 400 pp., 1986.

Cox, S., Cuthbert, A., Daisey, P., Davidson, J., Johnson, S., Keighan, E., Lake, R., Mabrouk, M., Margoulies, S., and Martell, R.: OpenGIS$^{®}$ Geography Markup Language (GML) Implemen-

10 tation Specification, version 2.1, available in http://www.opengis.org/techno/specs/02-009/GML2-11.html (last access: 29 June 2012), 2002.

Demsar, J., Zupan, B., Leban, G., and Curk, T.: Orange: From experimental machine learning to interactive data mining, European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Lecture Notes Comput. Sci., 3202/2004,

15 537–539, doi:10.1007/978-3-540-30116-5_58, 2004.

ESRI: ESRI Shapefile Technical Description [online], Available from http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf (last access: 8 May 2012), 1998.

Euler, L. and Sten, J.: Euler's original text (in Latin) and English translation [online], available from http://www.17centurymaths.com/contents/euler/e478tr.pdf, last access: 8 May 2012.

20 Geraci, A., Katki, F., McMonegal, L., Meyer, B., Lane, J., Wilson, P., Radatz, J., Yee, M., Porteous, H., and Springsteel, F.: IEEE Standard Computer Dictionary: Compilation of IEEE Standard Computer Glossaries, The Institute of Electrical and Electronics Engineers, New York, USA, 1991.

Greiner, B.: Euler rotations in plate-tectonic reconstructions, Comput. Geosci., 25, 209–216,

25 1999.

Gurnis, M., Turner, M., Zahirovic, S., Dicaprio, L., Spasojevich, S., Muller, R. D., Boyden, J., Seton, M., Manea, V. C., and Bower, D. J.: Plate Reconstructions with Continuously Closing Plates, Comput. Geosci., 35–42, doi:10.1016/j.cageo.2011.04.014, 2012.

Heim, M.: Exploring Indiana Highways: Trip Trivia, Exploring America's Highway, Travel Organi-

30 zation Network Exchange, Inc., Wabasha, 2007.

Hellinger, S. J.: The uncertainties of finite rotations in plate tectonics, J. Geophys. Res., 86, 9312–9318, 1981.

The GPlates
Geological
Information Model

X. Qin et al.

Title Page

Abstract | Introduction

Conclusions | References

Tables | Figures

◄◄ | ►►

◄ | ►

Back | Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion

Lake, R.: Geography mark-up language (GML), Wiley, 2004.

Lake, R.: The application of geography markup language (GML) to the geological sciences, Comput. Geosci., 31, 1081–1094, 2005.

Landgrebe, T. C. W. and Muller, R. D.: A Spatio-Temporal Knowledge-Discovery Platform for Earth-Science Data, Digital Image Computing Techniques and Applications (DICTA), 2011 International Conference on, 6–8 December 2011, 394–399, doi:10.1109/DICTA.2011.73, 2011.

Larman, C.: Applying UML and patterns: an introduction to object-oriented analysis and design and iterative development, Prentice Hall PTR, 2004.

Lee, Y. T.: Information modeling: From design to implementation, http://www.mel.nist.gov/msidlibrary/doc/tina99im.pdf, last access: 29 June 2012, 315–321, 1999.

Müller, R. D., Sdrolias, M., Gaina, C., and Roest, W. R.: Age, spreading rates and spreading asymmetry of the world's ocean crust, Geochem. Geophy. Geosy., 9, Q04006, doi:04010.01029/02007GC001743, 2008.

Panagiotis, A. V.: Web feature service (WFS) implementation specification, OGC document, 04-094, available from: http://www.opengeospatial.org/standards/wfs/ (last access: 29 June 2012), 2005.

Peng, Z. R. and Zhang, C.: The roles of geography markup language (GML), scalable vector graphics (SVG), and Web feature service (WFS) specifications in the development of Internet geographic information systems (GIS), J. Geogr. Syst., 6, 95–116, 2004.

Sen, M. and Duffy, T.: GeoSciML: Development of a generic GeoScience Markup Language, Comput. Geosci., 31, 1095–1103, 2005.

Simons, B., Boisvert, E., Brodaric, B., Cox, S., Duffy, T. R., Johnson, B. R., Laxton, J. L., and Richard, S.: GeoSciML: enabling the exchange of geological map data, ASEG Extended Abstracts, 1–4, 2006.

Tan, E., Leng, W., Zhong, S., and Gurnis, M.: CitcomS v3. 0-A compressible thermo-chemical mantle convection code, American Geophysical Union, Fall Meeting 2007, abstract #DI14A-01, 2007.

Wessel, P.: Seamounts, Geology, in: Encyclopedia of Islands, edited by: Clague, D. A. and Gillespie, R., UC Press, Berkeley, CA,, 821–825, 2009.

Wessel, P. and Smith, W. H. F.: New, improved version of Generic Mapping Tools released, EOS Trans. AGU, 579, 1998.

Title Page

Abstract | Introduction

Conclusions | References

Tables | Figures

|◄ | ►|

◄ | ►

Back | Close

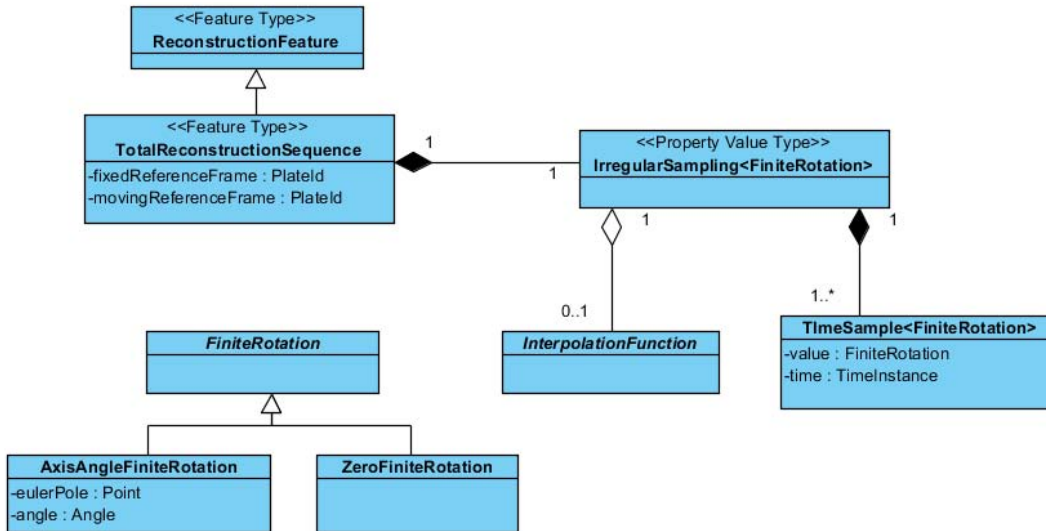Full Screen / Esc

Printer-friendly Version

Interactive Discussion

Williams, S. E., Muller, R. D., Landgrebe, T. C. W., and Whittaker, J. M.: An open-source software environment for visualizing and refining plate tectonic reconstructions using high-resolution geological and geophysical data sets, GSA Today, 22, 4–9, doi:10.1130/GSATG139A.1, 2012.

The GPlates Geological Information Model

X. Qin et al.

Title Page

Abstract | Introduction

Conclusions | References

Tables | Figures

◄◄ | ►►

◄ | ►

Back | Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion

Feature
-featureId : UniqueID
-revisionId : UniqueID

InstantaneousFeature
-reconstructedTime : TimeInstant

0..1

0..*

TimeVariantFeature
-validTime : TimePeriod

ReconstructionFeature

ReconstructableFeature
-reconstructionPlateId : PlateId

TopologicalFeature

TotalReconstructionSequence
-fixedReferenceFrame : PlateId
-fixedReferenceFrame2 : PlateId
-movingReferenceFrame : PlateId
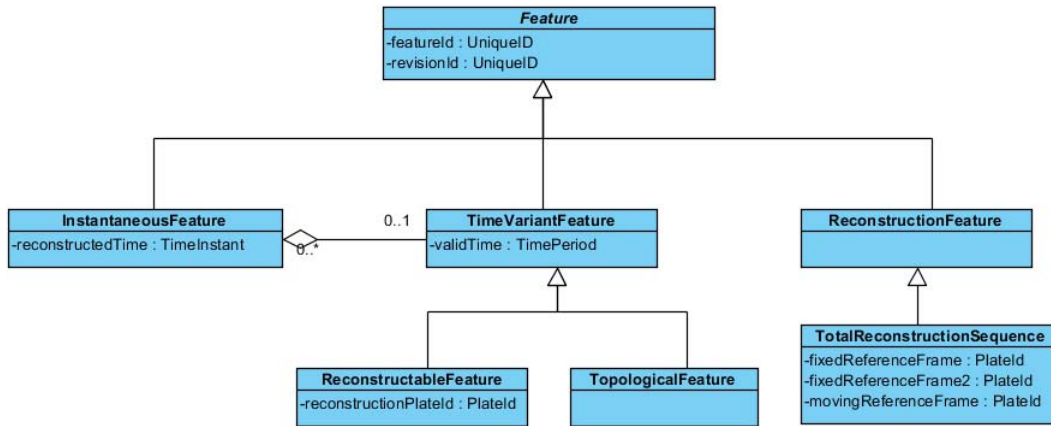
Fig. 1. UML diagram of GPGIM features.

Fig. 2. UML diagram of ReconstructionFeature and FiniteRotation.

ReconstructableFeature
- *TangibleFeature*
- *ArtificialFeature*
- *TopologicalBoundaryGuide*
- *UnclassifiedFeature*

Fig. 3. Four main subcategories of ReconstructableFeature in GPGIM.

TangibleFeature
- *AbstractField*
  - Bathymetry
  - Topography
  - Gravimetry
  - Magnetics
  - GlobalElevation
  - OceanicAge
  - CrustalThickness
  - DynamicTopography
  - MantleDensity
  - HeatFlow
  - SedimentThickness
  - Roughness
  - SpreadingRate
  - SpreadingAsymmetry
  - Stress
- *TectonicSection*
  - MidOceanRidge
  - ContinentalRift
  - SubductionZone
  - OrogenicBelt
  - Transform
  - FractureZone
  - PassiveContinentalBoundary
- *Isochron*
- *MagneticAnomalyIdentification*
- *MagneticAnomalyShipTrack*
- *FractureZoneIdentification*
- *AbstractGeologicalPlane*
  - GeologicalPlane
  - FoldPlane
  - AbstractGeologicalContact
    - Unconformity
    - TerraneBoundary
    - UnknownContact
    - Fault
- *AbstractRockUnit*
  - BasicRockUnit
- *Suture*
- *PseudoFault*
- *IslandArc*
- *HotSpotTrail*
- *HotSpot*
- *Seamount*
- *Volcano*
- *AseismicRidge*
- *Coastline*
- *Craton*
- *LargeIgneousProvince*
- *Basin*
- *ExtendedContinentalCrust*
- *TransitionalCrust*
- *ContinentalFragment*
- *GeologicalLineation*

Fig. 4. The complete list of tangible features in GPGIM.

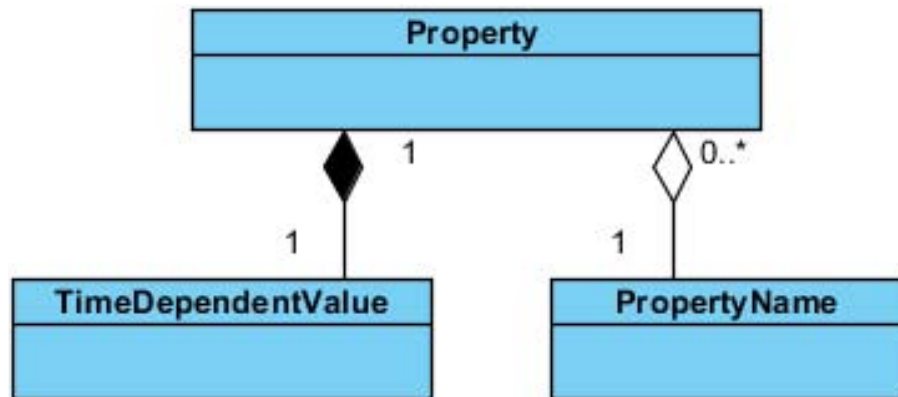**The GPlates Geological Information Model**

X. Qin et al.

Artificial Feature
- *ClosedPlateBoundary*
- *ClosedContinentalBoundary*
- *InferredPaleoBoundary*
- *OldPlatesGridMark*

Fig. 5. Four artificial features defined in GPGIM.

Fig. 6. UML diagram of feature property in GPGIM.

Title Page

Abstract | Introduction

Conclusions | References

Tables | Figures

Back | Close

Full Screen / Esc

Printer-friendly Version
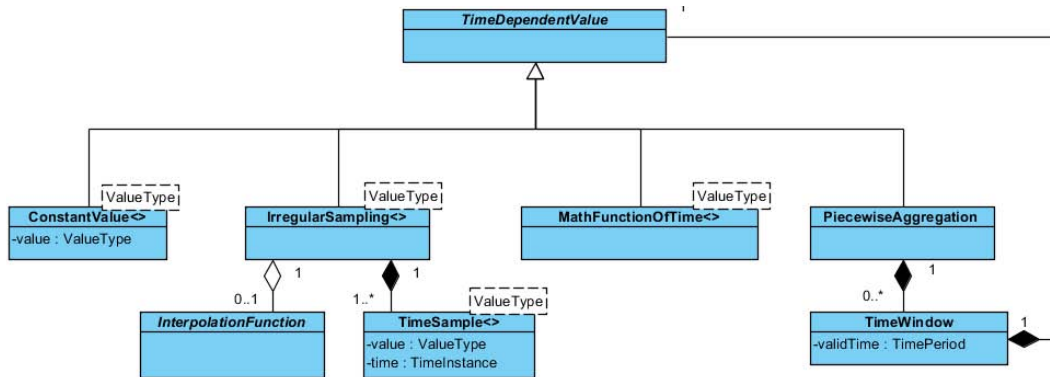
Interactive Discussion

Fig. 7. UML diagram of property value types in GPGIM.

```
<xs:element abstract="true" substitutionGroup="gml:AbstractFeature" type="gpml:AbstractFeatureType" name="AbstractFeature"/>
    <!-- ======================================================================= -->
    <!-- ======================================================================= -->
-   <xs:complexType abstract="true" name="AbstractFeatureType">
    -   <xs:complexContent>
        -   <xs:extension base="gml:AbstractFeatureType">
            -   <xs:sequence>
                    + <xs:element type="gpml:featureId" name="identity">
                    + <xs:element type="gpml:revisionId" name="revision">
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
```

Fig. 8. The AbstractFeature schema in GPML.

```xml
<xsd:element substitutionGroup="gml:AbstractFeatureCollection" type="wfs:FeatureCollectionType" name="FeatureCollection"/>
    <!-- ========================================================================= -->
- <xsd:complexType name="wfs:FeatureCollectionType">
  - <xsd:complexContent>
    - <xsd:extension base="gml:AbstractFeatureCollectionType">
        <xsd:attribute type="xsd:string" name="lockId" use="optional"/>
        <xsd:attribute type="xsd:dateTime" name="timeStamp" use="optional"/>
        <xsd:attribute type="xsd:nonNegativeInteger" name="numberOfFeatures" use="optional"/>
      </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
```

Fig. 9. The FeatureCollection schema in WFS.

```
- <complexType abstract="true" name="AbstractFeatureCollectionType">
    - <complexContent>
        - <extension base="gml:AbstractFeatureType">
            - <sequence>
                <element maxOccurs="unbounded" minOccurs="0" ref="gml:featureMember"/>
                <element minOccurs="0" ref="gml:featureMembers"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

Fig. 10. The AbstractFeatureCollection schema in GML.

```
<element type="gml:FeaturePropertyType" name="featureMember"/>
<element type="gml:FeaturePropertyType" name="featureProperty"/>
    <!-- ===================================================================== -->
- <complexType name="FeaturePropertyType">
  - <sequence minOccurs="0">
        <element ref="gml:AbstractFeature"/>
    </sequence>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
    <!-- ===================================================================== -->
- <complexType name="FeatureArrayPropertyType">
  - <sequence maxOccurs="unbounded" minOccurs="0">
        <element ref="gml:AbstractFeature"/>
    </sequence>
  </complexType>
```

Fig. 11. The featureMember schema in GML.

```
- <xs:complexType name="IsochronType">
  - <xs:complexContent>
    - <xs:extension base="gpml:TangibleFeatureType">
      - <xs:sequence>
        - <xs:element type="gpml:FeatureReferencePropertyType" name="conjugate" maxOccurs="unbounded" minOccurs="0">
          - <xs:annotation>
              <xs:documentation>A reference to the 'pair' or 'twin' of this isochron.</xs:documentation>
            </xs:annotation>
          </xs:element>
        - <xs:element type="gpml:plateId" name="conjugatePlateId" maxOccurs="unbounded" minOccurs="0">
          - <xs:annotation>
              <xs:documentation>A reference to the plate Id that has the 'pair' or 'twin' of this isochron.</xs:documentation>
            </xs:annotation>
          </xs:element>
        - <xs:element minOccurs="0" ref="unclassifiedGeometry">
          - <xs:annotation>
              <xs:documentation>All geometry defined for the feature where the purpose is not given or
                 unclear.</xs:documentation>
            </xs:annotation>
          </xs:element>
        - <xs:element minOccurs="0" ref="centerLineOf">
          - <xs:annotation>
              <xs:documentation>The central line defining the Isochron.</xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
```

Fig. 12. The definition of Isochron feature in GPML.

```
- <xs:complexType name="ConstantValueType">
    - <xs:complexContent>
        - <xs:extension base="gpml:TimeDependentPropertyValueType">
            - <xs:sequence>
                - <xs:element name="value">
                    - <xs:complexType mixed="true">
                        - <xs:sequence>
                            <xs:any minOccurs="0"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element type="xs:string" name="description" minOccurs="0"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

Fig. 13. The Constant Property Value schema in GPML.

```
- <xs:complexType name="IrregularSamplingType">
    - <xs:complexContent>
        - <xs:extension base="gpml:TimeDependentPropertyValueType">
            - <xs:sequence>
                  <xs:element type="gpml:TimeSamplePropertyType" name="timeSample" maxOccurs="unbounded"/>
                  <xs:element type="gpml:InterpolationFunctionPropertyType" name="interpolationFunction" minOccurs="0"/>
              </xs:sequence>
          </xs:extension>
      </xs:complexContent>
  </xs:complexType>
```

Fig. 14. The Irregular Sampling Value schema in GPML.

```
- <xs:complexType name="PiecewiseAggregrationType">
    - <xs:complexContent>
        - <xs:extension base="gpml:TimeDependentPropertyValueType">
            - <xs:sequence>
                - <xs:element name="timeWindow" maxOccurs="unbounded" minOccurs="0">
                    - <xs:complexType>
                        - <xs:sequence>
                            <xs:element type="gpml:TimeDependentPropertyValue" name="timeDependentPropertyValue"/>
                            <xs:element type="gml:TimePeriodType" name="validTime"/>
                            <xs:element type="xs:string" name="valueType"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
</xs:schema>
```

Fig. 15. The Piecewise Aggregation Value schema in GPML.

```
- <xs:element type="CenterLineOfType" name="centerLineOf" minOccurs="0">
    - <xs:annotation>
        <xs:documentation> Generally using gml:LineString, with the possibility of gml:MultiGeometry or gml:MultiCurve.
            </xs:documentation>
    </xs:annotation>
  </xs:element>
    <!-- ======================================================================= -->
    <!-- ======================================================================= -->
- <xs:complexType name="CenterLineOfType">
    - <xs:sequence>
        - <xs:element name="ConstantValue">
            - <xs:complexType>
                - <xs:complexContent>
                    - <xs:extension base="gpml:TimeDependentPropertyValueType">
                        - <xs:sequence>
                            - <xs:element name="value">
                                - <xs:complexType>
                                    - <xs:choice>
                                        <xs:element ref="gml:LineString"/>
                                        <xs:element ref="gml:MultiGeometry"/>
                                        <xs:element ref="gml:MultiCurve"/>
                                    </xs:choice>
                                </xs:complexType>
                            </xs:element>
                            <xs:element type="xs:string" name="description" minOccurs="0"/>
                        </xs:sequence>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
  </xs:complexType>
```

Fig. 16. The centerLine geometry schema in GPML.

```xml
- <xs:element type="OutlineOfType" name="outlineOf" minOccurs="0">
    - <xs:annotation>
        <xs:documentation> Geometry specifying a closed or partial outline. </xs:documentation>
      </xs:annotation>
  </xs:element>
    <!-- ======================================================================= -->
    <!-- ======================================================================= -->
- <xs:complexType name="OutlineOfType">
    - <xs:sequence>
        - <xs:element name="ConstantValue">
            - <xs:complexType>
                - <xs:complexContent>
                    - <xs:extension base="gpml:TimeDependentPropertyValueType">
                        - <xs:sequence>
                            - <xs:element name="value">
                                - <xs:complexType>
                                    - <xs:choice>
                                        <xs:element ref="gml:LineString"/>
                                        <xs:element ref="gml:MultiGeometry"/>
                                        <xs:element ref="gml:Polygon"/>
                                      </xs:choice>
                                  </xs:complexType>
                              </xs:element>
                              <xs:element type="xs:string" name="description" minOccurs="0"/>
                          </xs:sequence>
                      </xs:extension>
                  </xs:complexContent>
              </xs:complexType>
          </xs:element>
      </xs:sequence>
  </xs:complexType>
```

Fig. 17. The outline geometry schema in GPML.

```
- <xs:element type="ErrorBoundsType" name="errorBounds" minOccurs="0">
    - <xs:annotation>
        <xs:documentation> An optional error boundary for the feature. </xs:documentation>
    </xs:annotation>
  </xs:element>
    <!-- ========================================================================= -->
    <!-- ========================================================================= -->
- <xs:complexType name="ErrorBoundsType">
    - <xs:sequence>
        - <xs:element name="ConstantValue">
            - <xs:complexType>
                - <xs:complexContent>
                    - <xs:extension base="gpml:TimeDependentPropertyValueType">
                        - <xs:sequence>
                            <xs:element type="gml:AbstracGeometryType" name="value"/>
                            <xs:element type="xs:string" name="description" minOccurs="0"/>
                        </xs:sequence>
                    </xs:extension>
                </xs:complexContent>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
```

Fig. 18. The errorBounds geometry schema in GPML.

```
- <xs:element type="BoundaryType" name="boundary" minOccurs="0">
   - <xs:annotation>
        <xs:documentation> Represent as a Polygon. </xs:documentation>
   </xs:annotation>
 </xs:element>
   <!-- ========================================================================= -->
   <!-- ========================================================================= -->
- <xs:complexType name="BoundaryType">
   - <xs:sequence>
      - <xs:element name="ConstantValue">
         - <xs:complexType>
            - <xs:complexContent>
               - <xs:extension base="gpml:TimeDependentPropertyValueType">
                  - <xs:sequence>
                        <xs:element type="gml:PolygonType" name="value"/>
                        <xs:element type="xs:string" name="description" minOccurs="0"/>
                  </xs:sequence>
               </xs:extension>
            </xs:complexContent>
         </xs:complexType>
      </xs:element>
   </xs:sequence>
</xs:complexType>
```

Fig. 19. The boundary geometry schema in GPML.

```
- <xs:element type="PositionType" name="position" minOccurs="0">
  - <xs:annotation>
      <xs:documentation> lat/long point indicating the physical location </xs:documentation>
    </xs:annotation>
  </xs:element>
    <!-- ====================================================================== -->
    <!-- ====================================================================== -->
- <xs:complexType name="PositionType">
  - <xs:sequence>
    - <xs:element name="ConstantValue">
      - <xs:complexType>
        - <xs:complexContent>
          - <xs:extension base="gpml:TimeDependentPropertyValueType">
            - <xs:sequence>
                <xs:element type="gml:PointType" name="value"/>
                <xs:element type="xs:string" name="description" minOccurs="0"/>
              </xs:sequence>
            </xs:extension>
          </xs:complexContent>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
```

Fig. 20. The position schema in GPML.

```
- <xs:element type="UnclassifiedGeometryType" name="unclassifiedGeometry" minOccurs="0">
  - <xs:annotation>
      <xs:documentation> All geometry defined for the feature where the purpose is not given or unclear. </xs:documentation>
    </xs:annotation>
  </xs:element>
      <!-- ========================================================================= -->
      <!-- ========================================================================= -->
- <xs:complexType name="UnclassifiedGeometryType">
  - <xs:sequence>
    - <xs:element name="ConstantValue">
      - <xs:complexType>
        - <xs:complexContent>
          - <xs:extension base="gpml:TimeDependentPropertyValueType">
            - <xs:sequence>
                <xs:element type="gml:AbstracGeometryType" name="value"/>
                <xs:element type="xs:string" name="description" minOccurs="0"/>
              </xs:sequence>
            </xs:extension>
          </xs:complexContent>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
```

Fig. 21. The unclassifiedGeometry schema in GPML.

416

```
288      0.000    0.0000     0.0000     0.0000  291 ! FLI-CSS Falkland Islands-Colorado Subplate
288    175.000    0.0000     0.0000     0.0000  291 ! FLI-CSS Grunow et al 1991
288    175.000   47.4000   -33.4000    55.7700  701 ! FLI-AFR Grunow et al 1991 crossover
288    200.000   45.3000   169.2000  -156.2500  701 ! FLI-AFR Grunow et al 1991
288    600.000   45.3000   169.2000  -156.2500  701 ! FLI-AFR Grunow et al 1991
290      0.000    0.0000     0.0000     0.0000  202 ! SSS-PRB Salado Subplate-Parana Basin
290    126.700    0.0000     0.0000     0.0000  202 ! SSS-PRB Nuernberg & Mueller 1991
290    131.700  -32.2000   -64.5000     0.7300  202 ! SSS-PRB Nuernberg & Mueller 1991
290    149.500  -32.2000   -64.5000     1.1800  202 ! SSS-PRB Nuernberg & Mueller 1991
290    600.000  -32.2000   -64.5000     1.1800  202 ! SSS-PRB Nuernberg & Mueller 1991
```
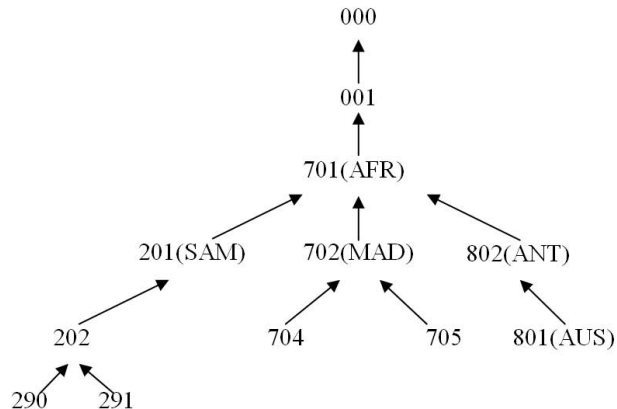
Fig. 22. A snippet of rotation file.

Fig. 23. An illustration of a reconstruction tree showing the relative rotation hierarchy of different tectonic plates to each other at a time instance in the past (AFR: Africa; ANT: Antarctica; AUS: Australia; MAD: Madagascar; SAM: South America).

The GPlates Geological Information Model

X. Qin et al.
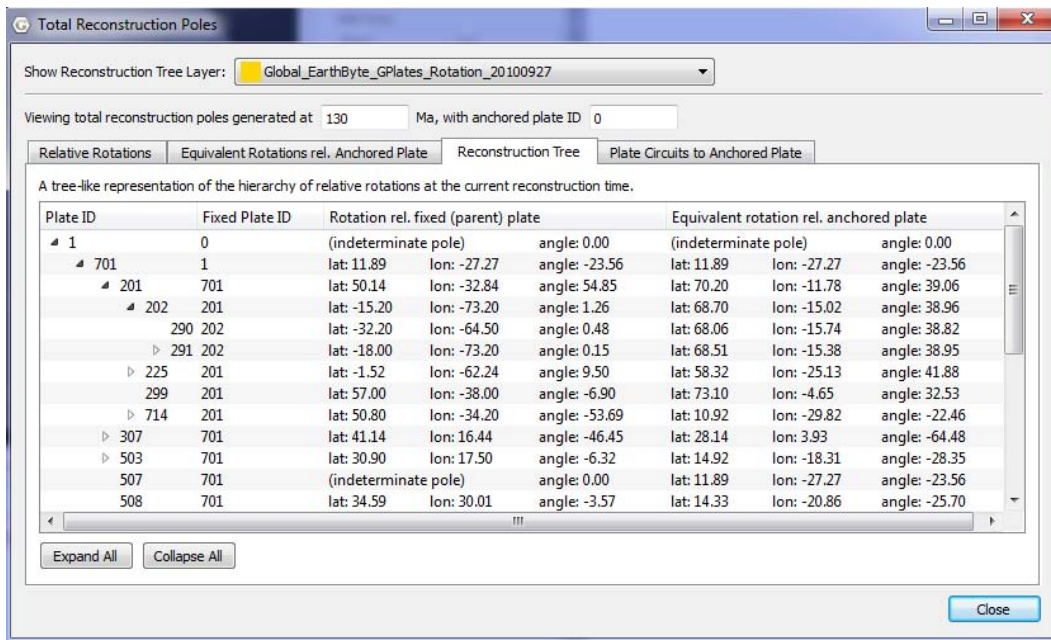
Fig. 24. The screenshot of GPlates reconstruction tree dialog.
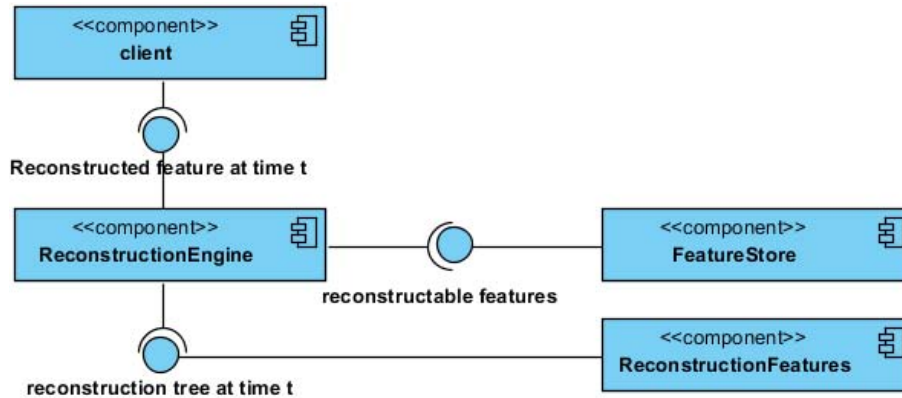
Fig. 25. Component diagram of reconstruction process.

Full Screen / Esc

Printer-friendly Version

Interactive Discussion

```
<!-- definition of feature "Raster"  -->
<xs:element name="Raster" type="gpml:RasterType" substitutionGroup="gml:_Feature" />
<!-- content model for feature "Raster"  -->
- <xs:complexType name="RasterType">
  - <xs:complexContent>
    - <xs:extension base="gpml:TimeVariantFeatureType">
      - <xs:sequence>
          <!-- Each file referenced by gml:File should have the same number and type of raster bands.  -->
        - <xs:element name="bandNames">
          - <xs:complexType>
            - <xs:sequence>
                <xs:element ref="gpml:RasterBandNames" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        - <xs:element name="domainSet">
          - <xs:complexType>
            - <xs:sequence>
                <!-- The 'valueType' property of ''TimeDependentPropertyValue'' must be 'AbstractGeoReferenceType'.  -->
                <xs:element ref="gpml:TimeDependentProperty" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        - <xs:element name="rangeSet">
          - <xs:complexType>
            - <xs:sequence>
                <!-- The 'valueType' property of ''TimeDependentPropertyValue'' must be 'gml:File'.  -->
                <xs:element ref="gpml:TimeDependentProperty" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <!-- Arbitrary name/value metadata associated with raster and its bands.  -->
        - <xs:element name="metadata">
          - <xs:complexType>
            - <xs:sequence>
                <!-- The 'valueType' property of ''TimeDependentPropertyValue'' must be 'RasterMetadata'.  -->
                <xs:element ref="gpml:TimeDependentProperty" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <!-- Coordinate reference system of raster.  -->
        - <xs:element name="coordinateReferenceSystem">
          - <xs:complexType>
            - <xs:sequence>
                <!-- The 'valueType' property of ''TimeDependentPropertyValue'' must be 'gml:AbstractCoordinateReferenceSystem'.  -->
                <xs:element ref="gpml:TimeDependentProperty" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
```

Fig. 26. Schema fragment for gpml:Raster feature.

421

```
<!-- definition of an affine geo-reference type  -->
<xs:element name="AffineGeoReference" type="gpml:AffineGeoReferenceType" substitutionGroup="gpml:AbstractGeoReference" />
<!--  content model for affine geo-reference type  -->
- <xs:complexType name="AffineGeoReferenceType">
  - <xs:complexContent>
    - <xs:extension base="gpml:AbstractGeoReferenceType">
      - <xs:sequence>
        - <xs:element name="rectifiedGrid">
          - <xs:complexType>
            - <xs:sequence>
                <xs:element ref="gml:RectifiedGrid" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!-- definition of a ground control points geo-reference type  -->
<xs:element name="GCPGeoReference" type="gpml:GCPGeoReferenceType" substitutionGroup="gpml:AbstractGeoReference" />
<!--  content model for ground control points geo-reference type  -->
- <xs:complexType name="GCPGeoReferenceType">
  - <xs:complexContent>
    - <xs:extension base="gpml:AbstractGeoReferenceType">
      - <xs:sequence>
        - <xs:element name="groundControlPoint" maxOccurs="unbounded">
          - <xs:complexType>
            - <xs:sequence>
              - <xs:element name="GroundControlPoint">
                  <!--  Associates position in raster image with location in georeferenced coordinates.  -->
                - <xs:complexType>
                  - <xs:sequence>
                      <xs:element name="pixelX" type="xs:double" />
                      <xs:element name="pixelY" type="xs:double" />
                      <xs:element name="georeferencedX" type="xs:double" />
                      <xs:element name="georeferencedY" type="xs:double" />
                      <!--  Elevation of GCP is optional.  -->
                      <xs:element name="georeferencedZ" type="xs:double" minOccurs="0" />
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

Fig. 27. Schema fragments for raster georeferencing.

```
<!-- definition of raster metadata -->
<xs:element name="RasterMetadata" type="gpml:RasterMetadataType" />
<!-- content model raster metadata -->
- <xs:complexType name="RasterMetadataType">
  - <xs:sequence>
      <!-- Metadata for all raster bands. -->
    - <xs:element name="domains">
      - <xs:complexType>
        - <xs:sequence>
            <xs:element ref="gpml:RasterMetadataDomains" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <!-- Each raster band can also have its own metadata. -->
    - <xs:element name="bandDomains" minOccurs="0" maxOccurs="unbounded">
      - <xs:complexType>
        - <xs:sequence>
          - <xs:element name="BandDomains">
            - <xs:complexType>
              - <xs:sequence>
                  <xs:element name="bandName" type="xs:string" />
                - <xs:element name="domains">
                  - <xs:complexType>
                    - <xs:sequence>
                        <xs:element ref="gpml:RasterMetadataDomains" />
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
```
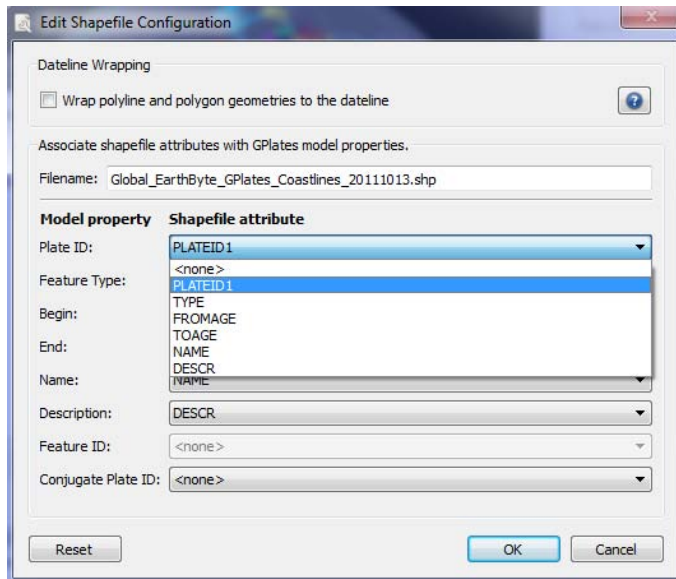
Fig. 28. Schema fragment for raster metadata.

```
<!-- definition of raster metadata domains -->
<xs:element name="RasterMetadataDomains" type="gpml:RasterMetadataDomainsType" />
<!-- content model raster metadata domains -->
- <xs:complexType name="RasterMetadataDomainsType">
  - <xs:sequence>
      <!-- Default domain (empty domain name). -->
    - <xs:element name="defaultDomainDictionary" minOccurs="0">
      - <xs:complexType>
        - <xs:sequence>
            <xs:element ref="gpml:RasterMetadataDictionary" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <!-- Non-default domains (non-empty domain name). -->
    - <xs:element name="domain" minOccurs="0" maxOccurs="unbounded">
      - <xs:complexType>
        - <xs:sequence>
          - <xs:element name="Domain">
            - <xs:complexType>
              - <xs:sequence>
                  <!-- The name of this domain. -->
                  <xs:element name="name" type="xs:string" />
                - <xs:element name="dictionary">
                  - <xs:complexType>
                    - <xs:sequence>
                        <xs:element ref="gpml:RasterMetadataDictionary" />
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
```

Fig. 29. Schema fragment for raster metadata domains.

Fig. 30. The shapefile attribute mapping dialog of GPlates.

```
- <GPlatesShapefileMap version="1">
     <Begin>FROMAGE</Begin>
     <Description>DESCR</Description>
     <End>TOAGE</End>
     <FeatureType>TYPE</FeatureType>
     <Name>NAME</Name>
     <ReconstructionPlateId>PLATEID1</ReconstructionPlateId>
  </GPlatesShapefileMap>
```

Fig. 31. The XML file keeping the mapping configuration.

Attribute name	Type;length	Description/Comment
PlateID1	short; 4	Unique plate identifier
FromAge	float; 8,3	Age of appearance
ToAge	float; 8,3	Age of disappearance
FeatureType	text; 150	Follow GPlates GPGIM
FeatureName	text; 250	Name of the feature
FeatureDescr	text; 250	Verbose description
DOI Ref	varchar	Unique DOI or verbose reference
Commen	text; 250	Comments on feature
Mod Author	varchar	Modification details
StandardV	varchar	Version of the standard

Fig. 32. Basic shapefile standard attribute names, types and description which allow direct interoperability between GIS shapefiles and GPGIM.

Back   Close

Full Screen / Esc
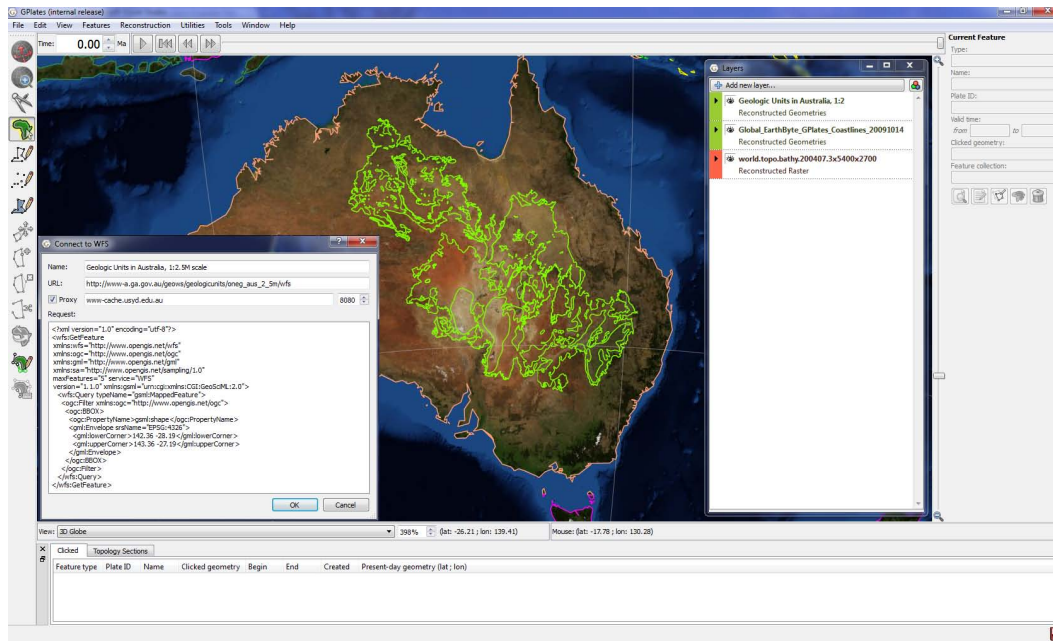
Printer-friendly Version

Interactive Discussion

Fig. 33. Screenshot showing GPlates displaying geometries (green) defined in GeoSciML data which is retrieved by Web Feature Service (WFS) from a Geosciences Australia website. The service base URL is http://www-a.ga.gov.au/geows/geologicunits/oneg_aus_2_5m/wfs. The WFS request is shown in the dialog (left). All GPlates layers are listed in the dialog (right).