GISI.pm: Object-Oriented Framework to Access Spatial Data from Perl Scripts

Alexandre Sorokine, Kurt Ackermann
Regional Science Institute
4-13 Kita-24 Nishi-2 Kita-ku
Sapporo 001-0024 Japan
Phone +81-11-717-6660
Fax +81-11-757-3610
`(sorokin|kurt)@vtt.co.jp`

2000 June 6, Munich, Germany

What is GISI.pm

GISI.pm is an object-oriented API (Application Programming Interface) for accessing spatial data from Perl scripts. GISI.pm is intended to define a set of classes and conventions to provide a consistent interface to access spatial data independent of underlying format or access method. GISI.pm is based on the OpenGIS Abstract specification Version 4.

Regional Science Institute

Architecture

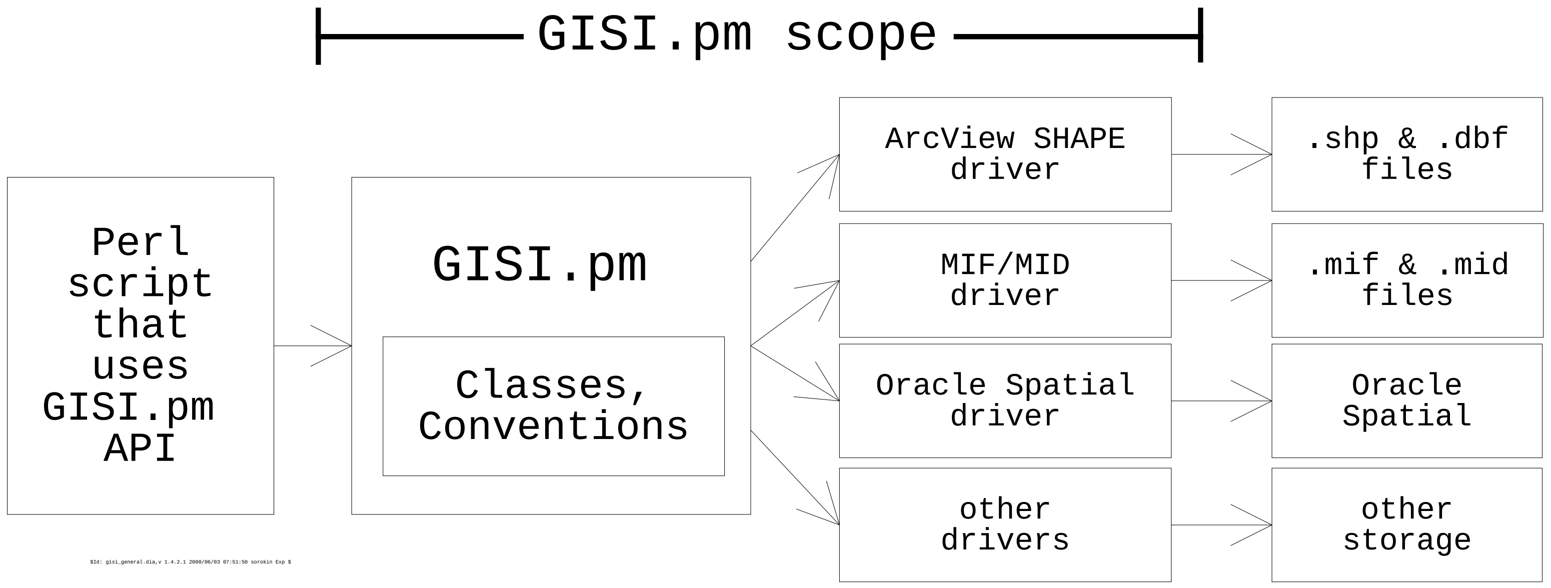

Figure 1: Architecture of a GISI.pm application

History

- Goal: reduce costs of casual GIS programming tasks
- Based on a variety of earlier developed scripts
- Approach: unify data access layer for existing scripts

Regional Science Institute

Scripting and System Programming Languages

- Major scripting languages
  - Perl
  - Python
  - Tcl/Tk
  - Visual Basic
- System programming languages
  - C
  - C++
  - Java

Regional Science Institute

Recommended Reading

Ousterhout, John K.

Scripting: Higher Level Programming for the 21st Century

IEEE Computer, 1998, vol. 31, #3 (March), pp. 23-30

`http://www.scriptics.com/people/john.ousterhout/scripting.html`

Regional Science Institute

Pros and Cons of Scripting Languages

- ✔ Increased pace of application development
- ✔ Reduced costs of application development
- ✔ Vast gluing capabilities
  - ➠ hybrid applications: C or C++ and Tcl/Tk, Perl, Python
- ✘ Reduced application speed

Scripting in GIS Applications

- ✔ ARC/INFO AML
- ✔ ArcView Avenue
- ✔ MapBasic
- ✔ TK GRASS

Gluing Capabilities of Perl

- Interfaces to DBMS (Perl DBI, oraperl, etc.)
- HTML or XML generators and parsers
- Interfaces to web servers (apache)
- Network protocols (HTTP, SMTP, etc.)
- User interfaces (TkPerl, Motif, Qt)
- DCEs (CORBA, OLE/COM)

Regional Science Institute

Perl DBI-like API for GIS

Goal:

- Develop a framework of object-oriented modules for Perl that is designed to provide uniform access to various sources of spatial data from Perl scripts

Regional Science Institute

Data Model Problems

1. No common data model in GIS:

   JDBC or the Perl DBI are built around relational databases that is

   - standardized
   - well studied
   - widely accepted

2. No direct mapping data model into language structures (e.g. geometry or topology)

3. Several representations of the same data possible

Solutions

- Use OpenGIS Abstract Specification as a common spatial data model
- Develop a hierarchy of abstract classes to represent the spatial data model

Why OpenGIS Abstract Specification:

- Intended to be a glue between a variety of GIS applications
    - this meshes well with the purpose of scripting languages
- Recognized and supported by the industry
- Many products are expected to comply
    - commonalities between the data models of the compliant products and our framework will simplify driver creation
- Demonstrates healthy progress and continuous updating
- Specification breadth is unrealistic to achieve for a limited group of developers

Regional Science Institute

GISI.pm Elements

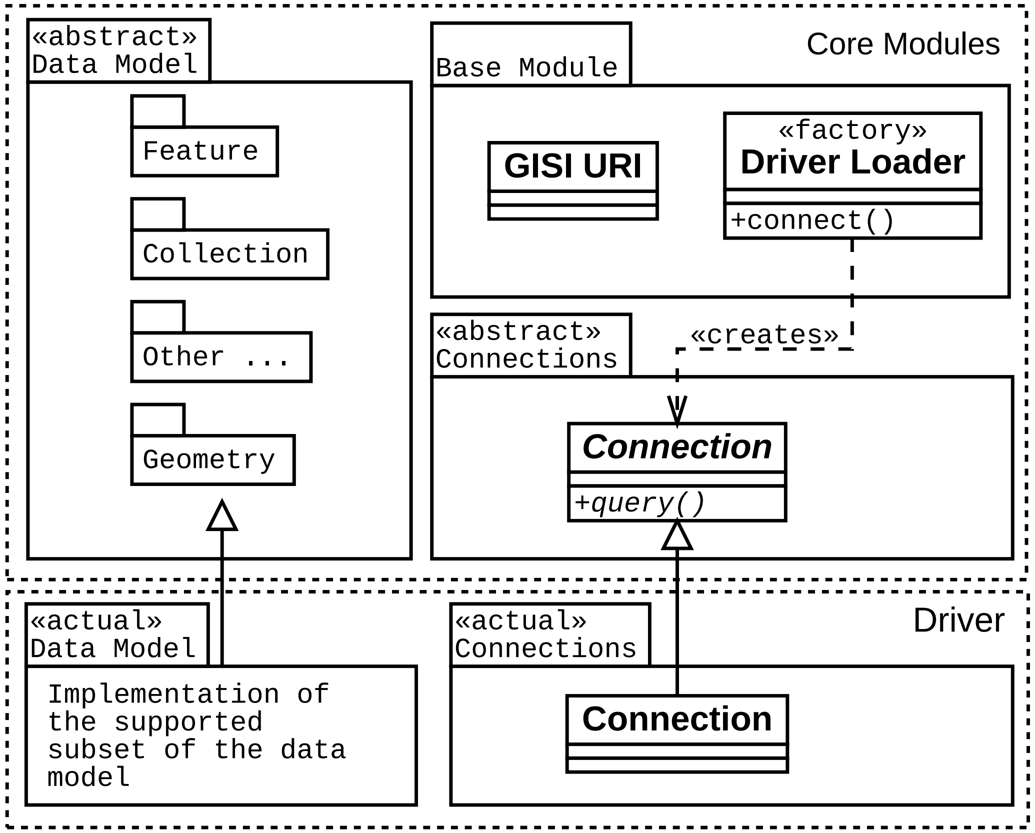

Figure 2: GISI.pm elements

Regional Science Institute

Interaction

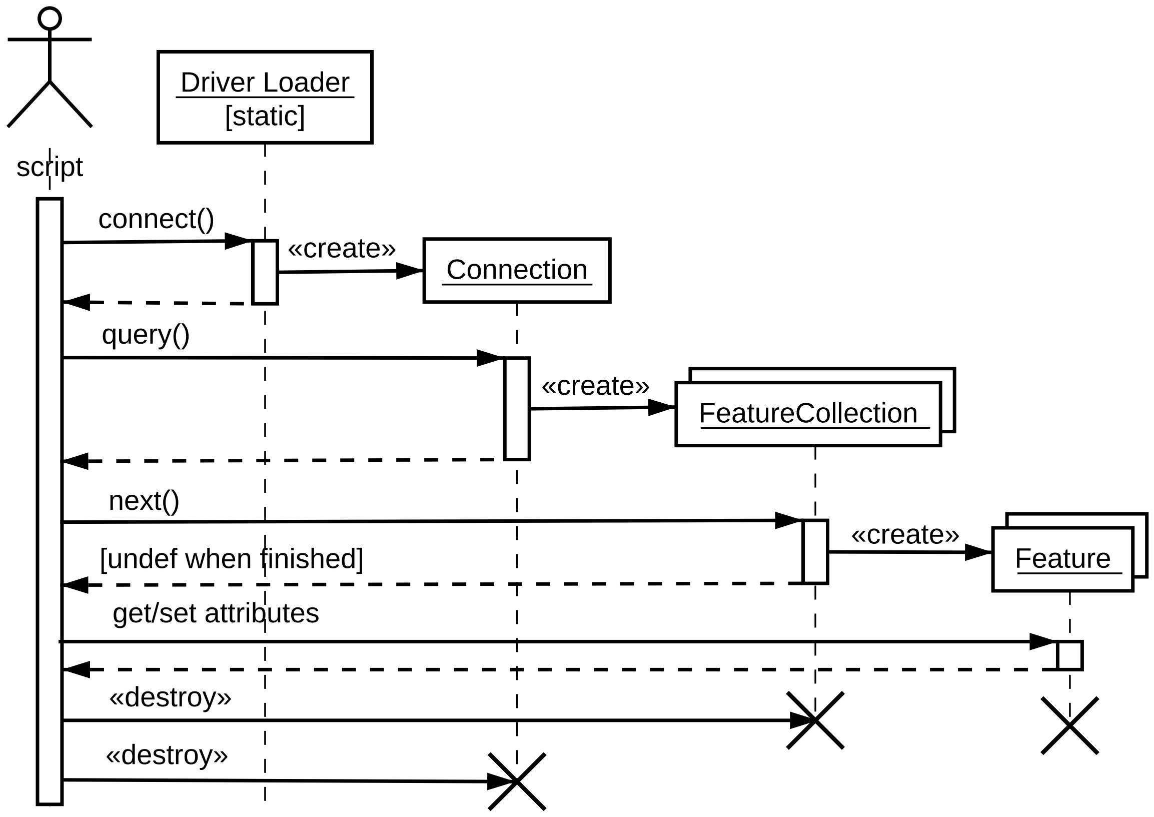

Figure 3: Interaction diagram

Feature and FeatureCollection

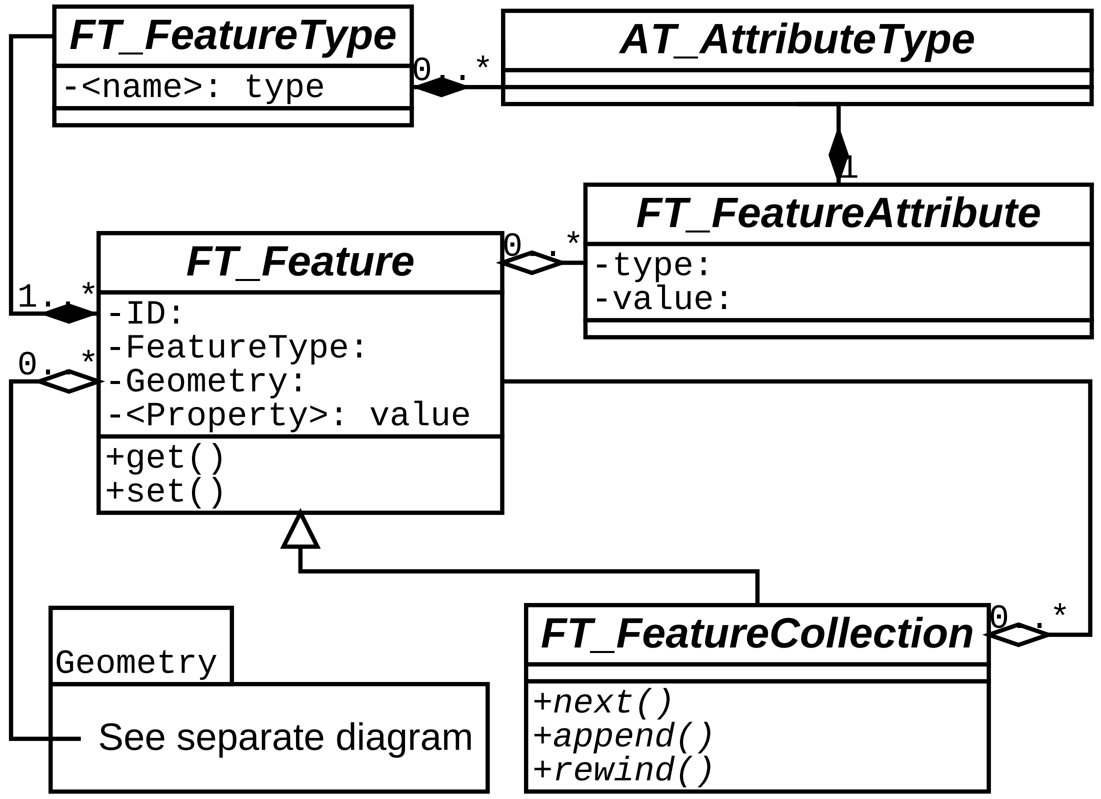

Figure 4: Supported subset (simplified)

Regional Science Institute

Geometry

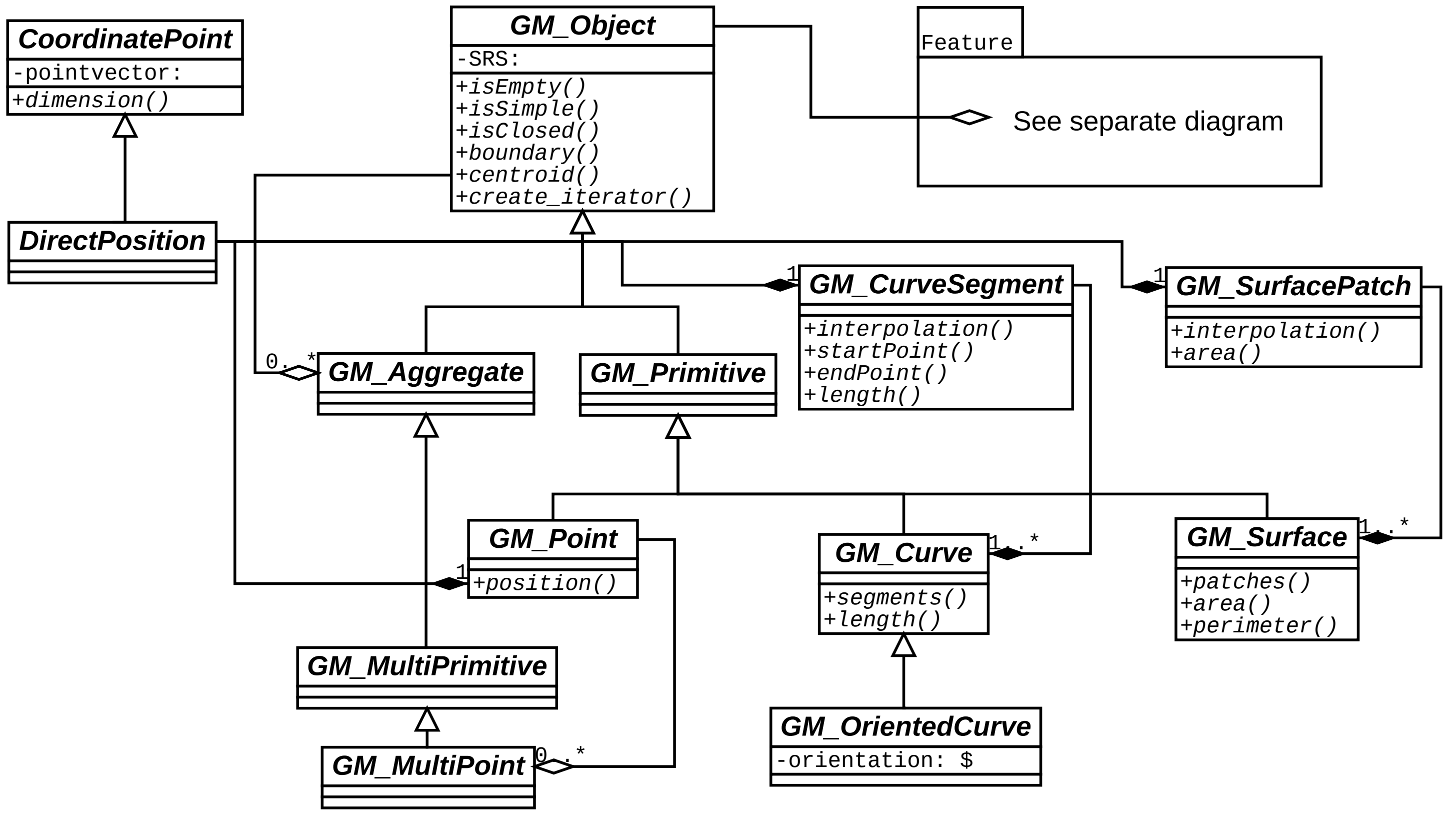

Figure 5: Supported subset (simplified)

Regional Science Institute

Additional Requirements:

- Keep GISI.pm thin and limit its scope to data access
- Preserve the maximum of information that comes with a dataset
  - geometry
  - attributes
  - metadata
  - etc. as defined in the specification.

Regional Science Institute

Drivers

- Can be implementation in Perl or C
- To interface Perl to C use of interface generators is possible (i.e. SWIG)
- Drivers may access either files or database servers
- Usually C drivers have 2 layers:

1. convert into raw Perl data structure
2. convert raw Perl data structures into GISI.pm objects

Regional Science Institute

Existing Drivers of General Interest (May 2000):

- ArcView SHAPE
- Several ARC/INFO `generate` command formats,
- MapInfo MIF/MID and

Drivers for tabular data

- CSV (comma separated value)
- DBASE

Regional Science Institute

Perl as OO Language

- Originally Perl was not object-oriented
- OO features appeared in version 5 in 1993.
- OO features naturally combine with a non-object-oriented style.
- OO features are not a part of the “core language”
  - built of elements that were already present in Perl
- Objects are just references associated with methods from an arbitrary module

Regional Science Institute

Lacking OO Features in Perl

- ✔ Abstract and static classes
- ✔ Protected and final methods or variables
- ✔ All constructs can be simply achieved by the development team following certain conventions
- ✔ This corresponds very well to Perl philosophy
- ✔ For example, private methods have to be implemented by just not mentioning them in the documentation

Pros and Cons of Perl OO Model

- In situations where the speed of development outweighs issues of code cleanliness, lack of restrictions may confer advantage

- Bigger and heavily object-oriented systems like GISI.pm have to be created very cautiously and strictly following the previously agreed conventions.

Regional Science Institute

Why Perl

- Language worth mentioning: Tcl/Tk, Python, Perl
- Reason for Choosing Perl:

1. team experience and existing code
2. plenitude of readily available language modules:
   - feature for automated modules download and installation
   - Perl Database independent interface
   - Tie::IxHash — mixture of a hash table and an array
   - standard URI
   - FreezeThaw – support for Perl serialized objects.

GISI.pm and Perl Modules

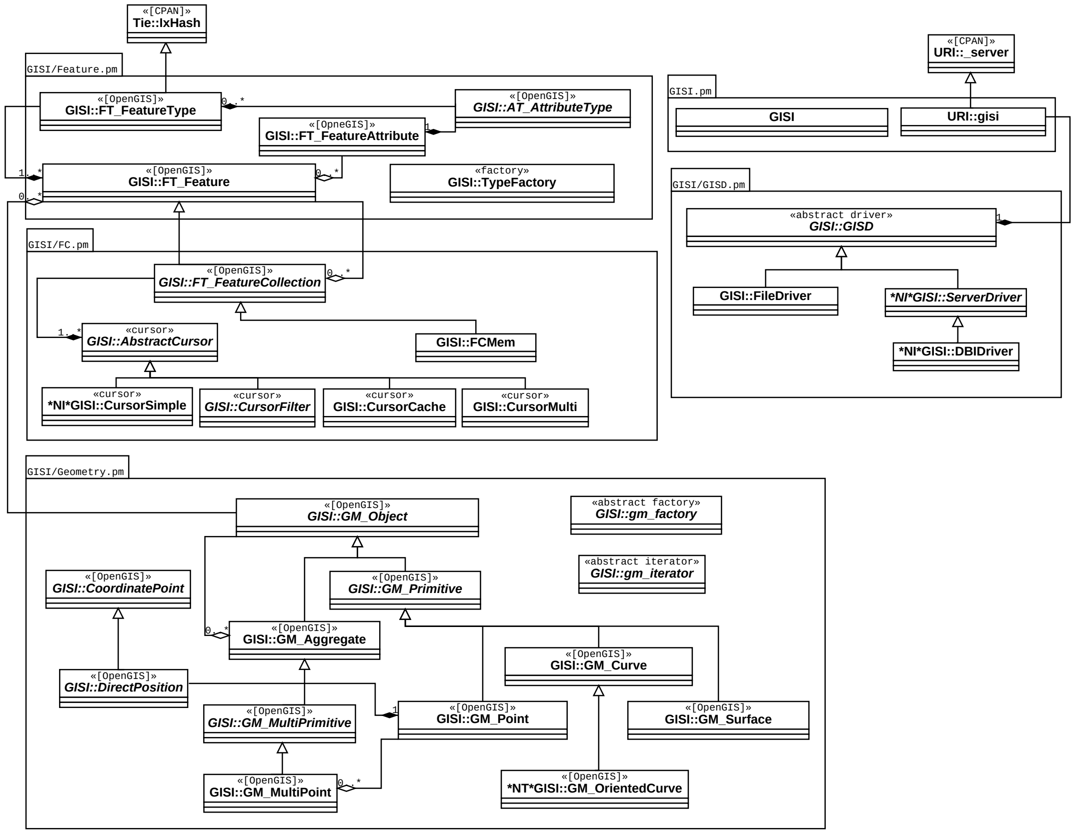

Figure 6: GISI.pm API organization in Perl modules

Regional Science Institute

Attribute Types

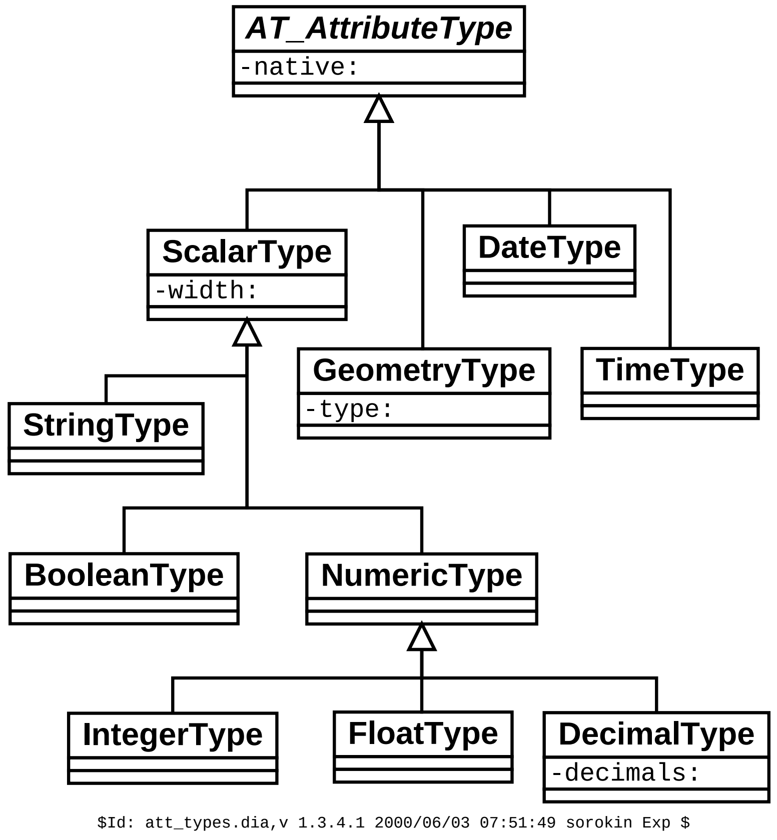

Figure 7: Attribute types

Regional Science Institute

Geometry Iterators

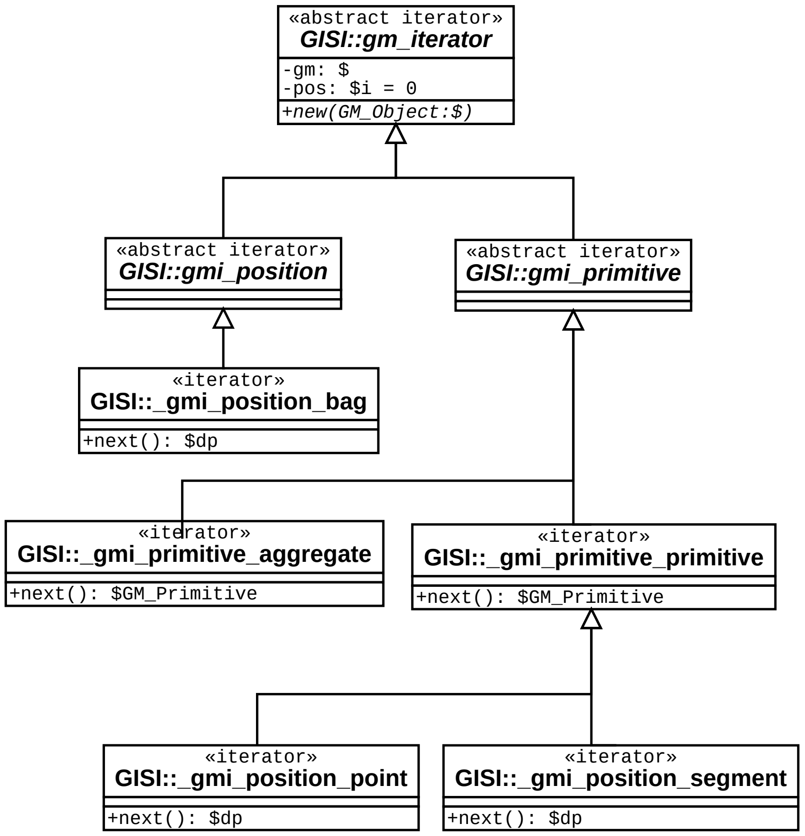

Figure 8: Geometry iterators

Regional Science Institute

Geometry Factories

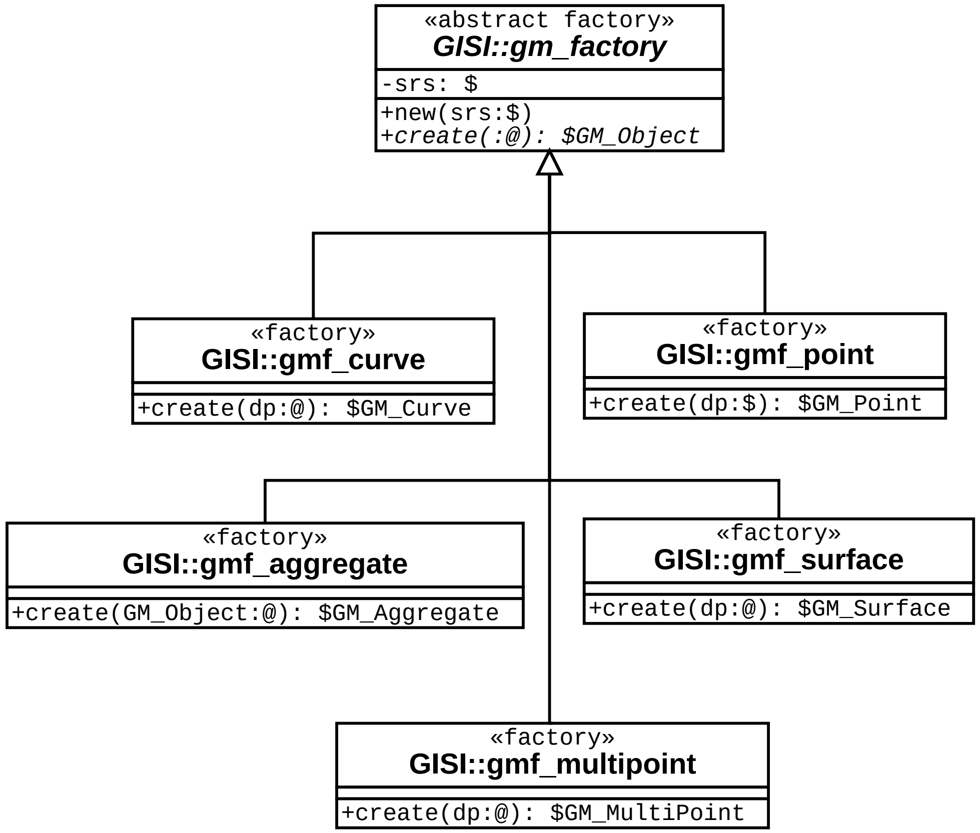

Figure 9: Geometry Factories

Regional Science Institute

Cursors

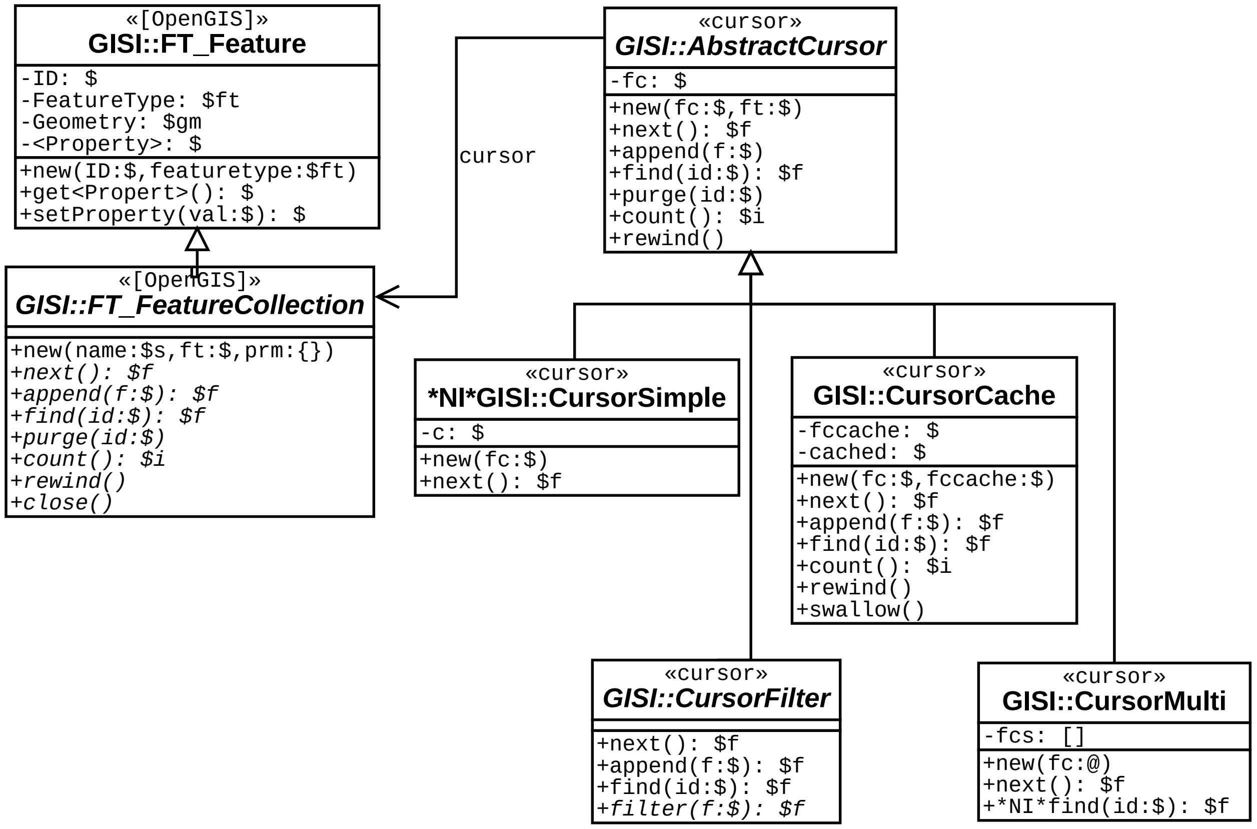

Figure 10: Cursors

Regional Science Institute

Testing

- Strong test suit
- Testing is important since because most errors expose themselves only during run-time
- Tests:

1. element test: set of tests for each method of each module
2. connectivity test: interaction of modules and drivers

Regional Science Institute

Connectivity Test

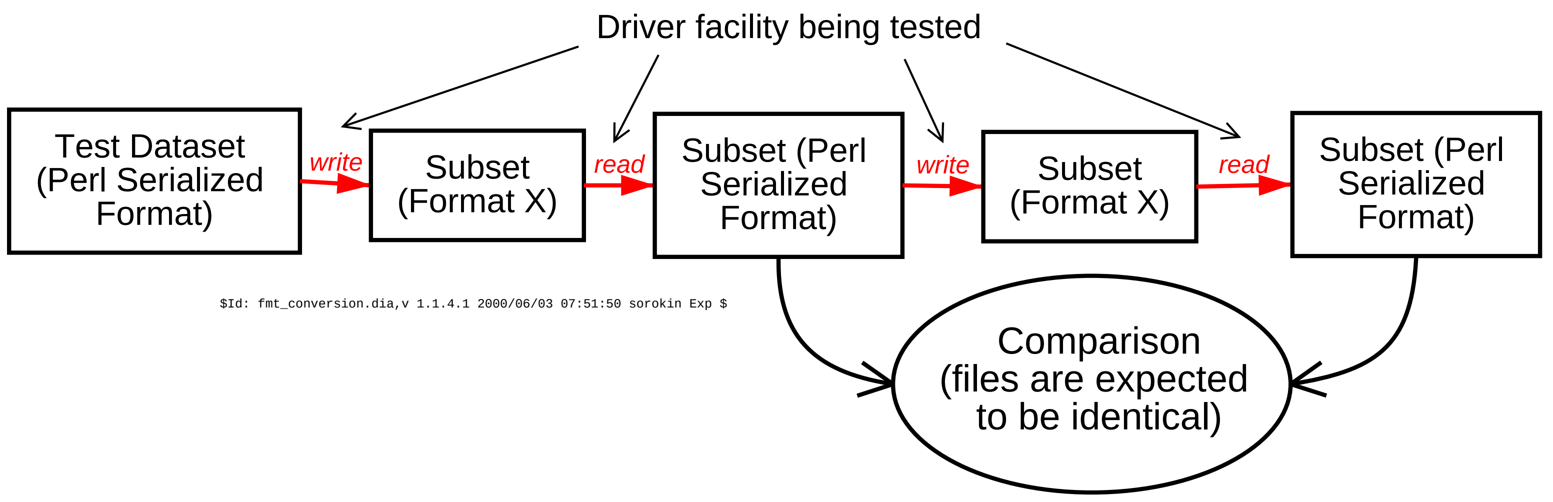
Driver facility being tested
Test Dataset (Perl Serialized Format)
write
Subset (Format X)
read
Subset (Perl Serialized Format)
write
Subset (Format X)
read
Subset (Perl Serialized Format)
$Id: fmt_conversion.dia,v 1.1.4.1 2000/06/03 07:51:50 sorokin Exp $
Comparison (files are expected to be identical)

Figure 11: Format conversion test

Regional Science Institute

Application Areas

1. Conversion of rarely used and unique formats
2. Correction of improperly formatted files
3. Conversion of commonly used file formats
4. Exploration of and collecting statistics on the datasets without converting them into any particular package
5. Custom geometric transformations and projection conversion without using commercial packages (thanks to PROJ.4)

Regional Science Institute

Shortcomings

- Insufficient error handling
- Lack of support for concurrent access
- Lack of topology
- No metadata support
- No spatial reference systems
- The biggest missing element is the coverage and gridded coverage
- Identifiers are only partially implemented

Regional Science Institute

Targets for Beta Version

1. Topology support
2. Spatial reference systems
3. Metadata
4. Feature identifiers
5. Gridded coverages
6. More drivers:
   (a) universal PerlDBI driver
   (b) non-spatial vector formats like SVG

Regional Science Institute

Feature Collections Problem

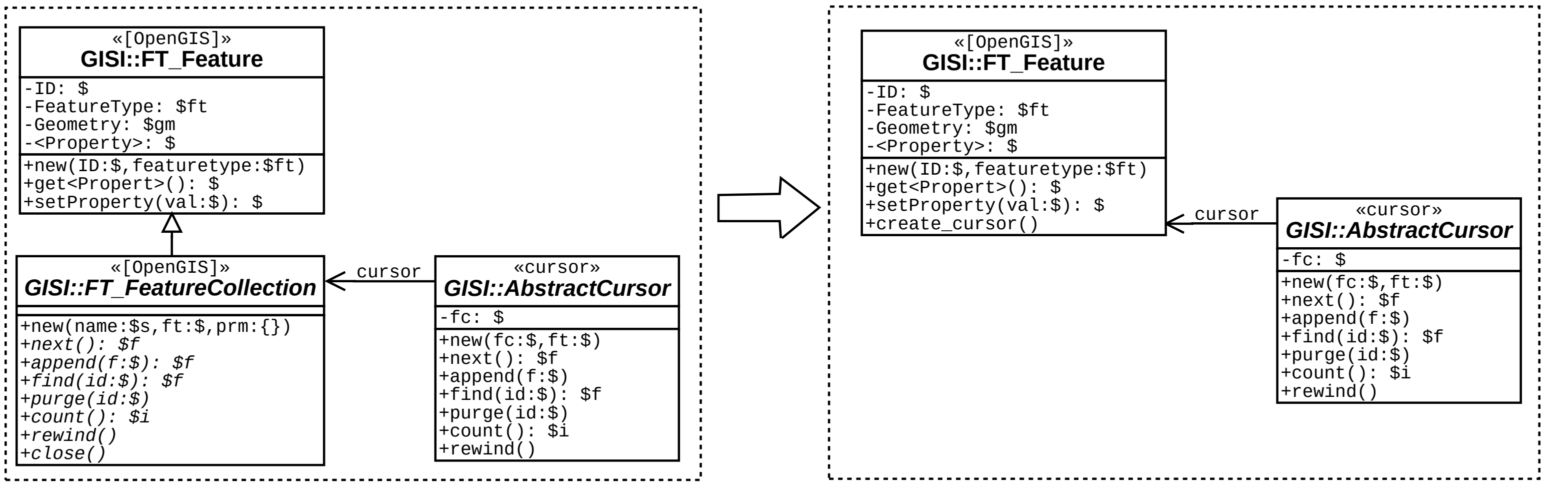

Figure 12: Changes to feature collection class

Regional Science Institute

Conclusions

1. GISI.pm works and is effective
2. GISI.pm are not as slow as was originally expected and enhancement is possible
3. Geometry specification has made remarkable progress

We think that scripting in GIS has a bright future for componentized, glue-oriented applications and prototyping. Frameworks similar to GISI.pm, if developed for a variety of scripting languages, can be used to glue together various applications and components: data access and storage systems, simulation models, computationally intensive algorithms, web and user interface applications.

Regional Science Institute

$Id: slides-2000-06.lyx,v 1.2.2.6 2000/06/20 08:19:30 sorokin Exp $