A Process and Environment for Embedding The R Software into TerraLib

Pedro Ribeiro de Andrade Neto
Paulo Justiniano Ribeiro Junior

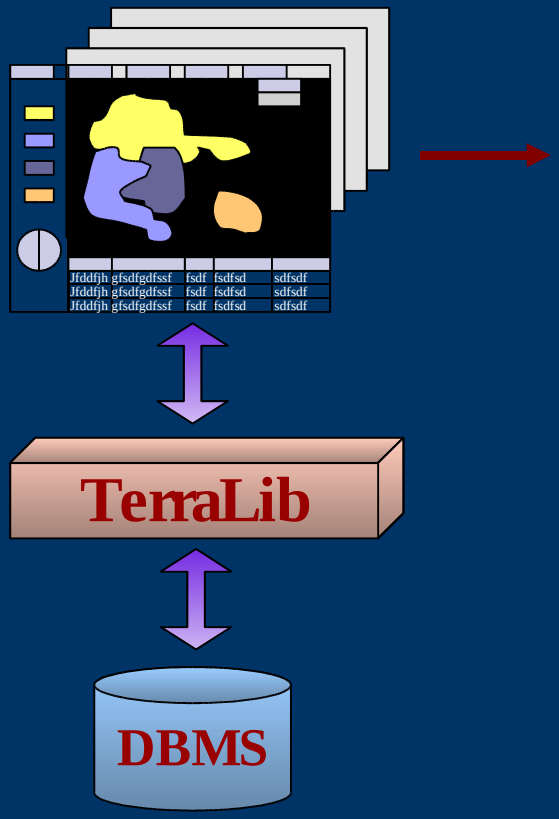
GeoInfo 2005

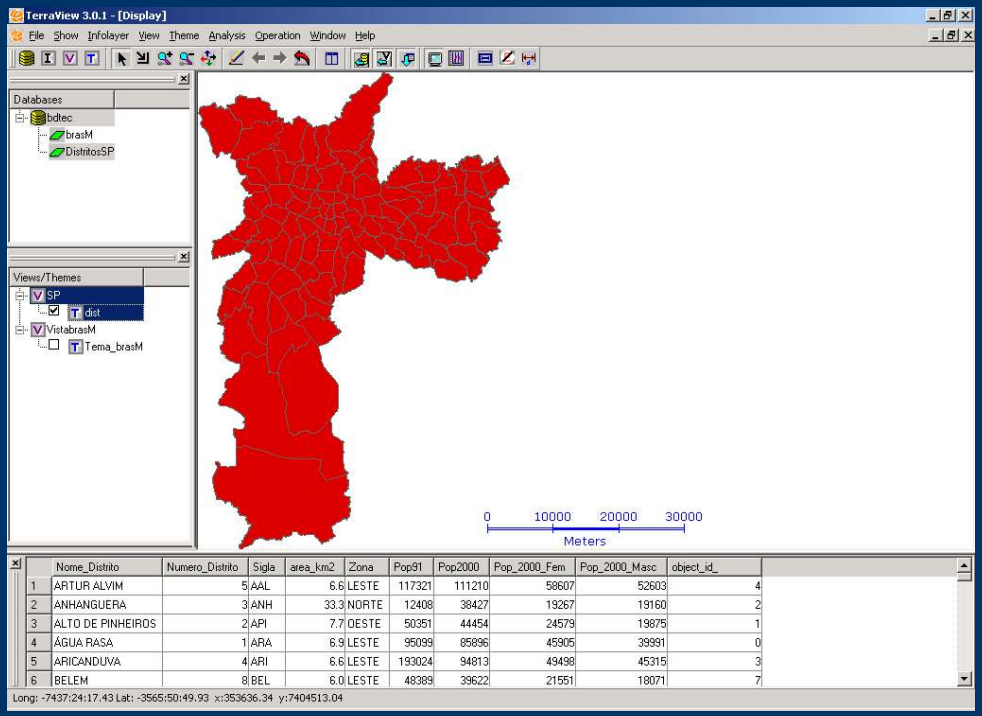
Outline

- Introduction
- Related Works
- aRT
- Process for Integration
- Conclusions

TerraLib

- Library for building *customized* GIS
- For programmers
- Needs specialized statistics analysis





R Project for Statistical Computing

- Free software

- Point pattern analysis: spatstat, splancs
- Geostatistics: gstat, geoR, geoRglm
- Areal data analysis: DCluster, spdep

- 32 packages currently available in the *Spatial Task View* on the CRAN repository

- Shared library (.dll, .so)

Integration

- Coupling R inside TerraLib
- Can benefit from specialized spatial statistical methods:
  - Point pattern analysis
  - Geostatistics
  - Areal data analysis

Objective

Propose an efficient way to incorporate R analysis scripts into TerraLib

- Use R as an evolving library for building personalized TerraLib based GIS

- Meet the needs of three agents:
  - Statisticians
  - TerraLib users (programmers)
  - GIS users

Related Works for R Users

R packages to access GIS data/functionalities:

- GRASS (Bivand and Netler 2000)
- RArcInfo (Gómes-Rubio 2005)
- rgdal (Keith and Bivand 2004)
- aRT (Andrade Neto et. al. 2005)

Problem:

- They require R knowledge, and then the target is *statisticians/R users*

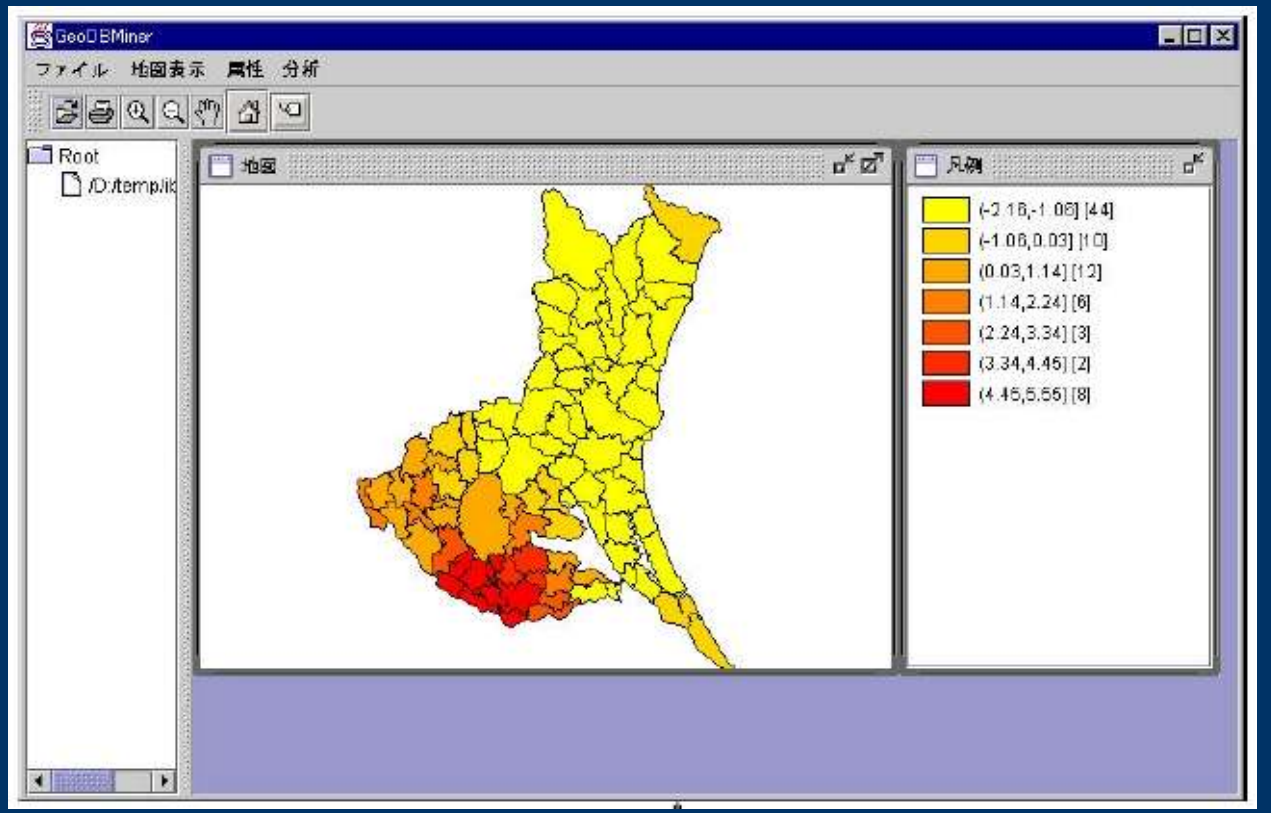
Example of an R Script

```
conn       = openConn(user="pedro")
db         = openDb(conn, "dbname")
thpoints   = openTheme(db, "themepoints")
thborder   = openTheme(db, "themeborder")

points = getPoints(thpoints)
border = getPolygons(thborder)
data   = getData(thpoints)

raster = krige(points, border, data)

l = createLayer(db, "lraster")
addRaster(l, raster)
thraster = createTheme(l, "themeraster")
```

Related Works for GIS Users

Graphic interfaces to R:

- Ono et. al. 2003
- Tait et. al. 2004



Problems:

- They describe the functionality, but not the proccess to aggregate new functionalities
- They exchange information with R using files

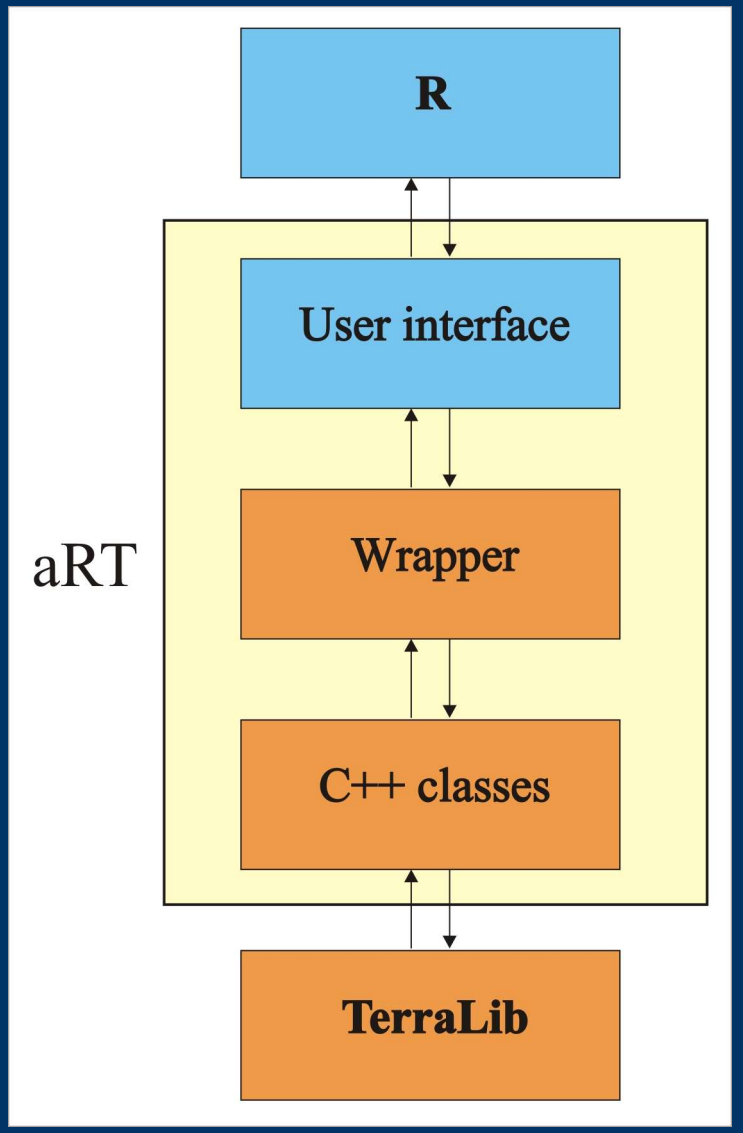
Related Works for GIS Programmers

No works...

The aRT package: R-TerraLib API

- R package that can connect to TerraLib databases
- Scripts for statistical analysis of data

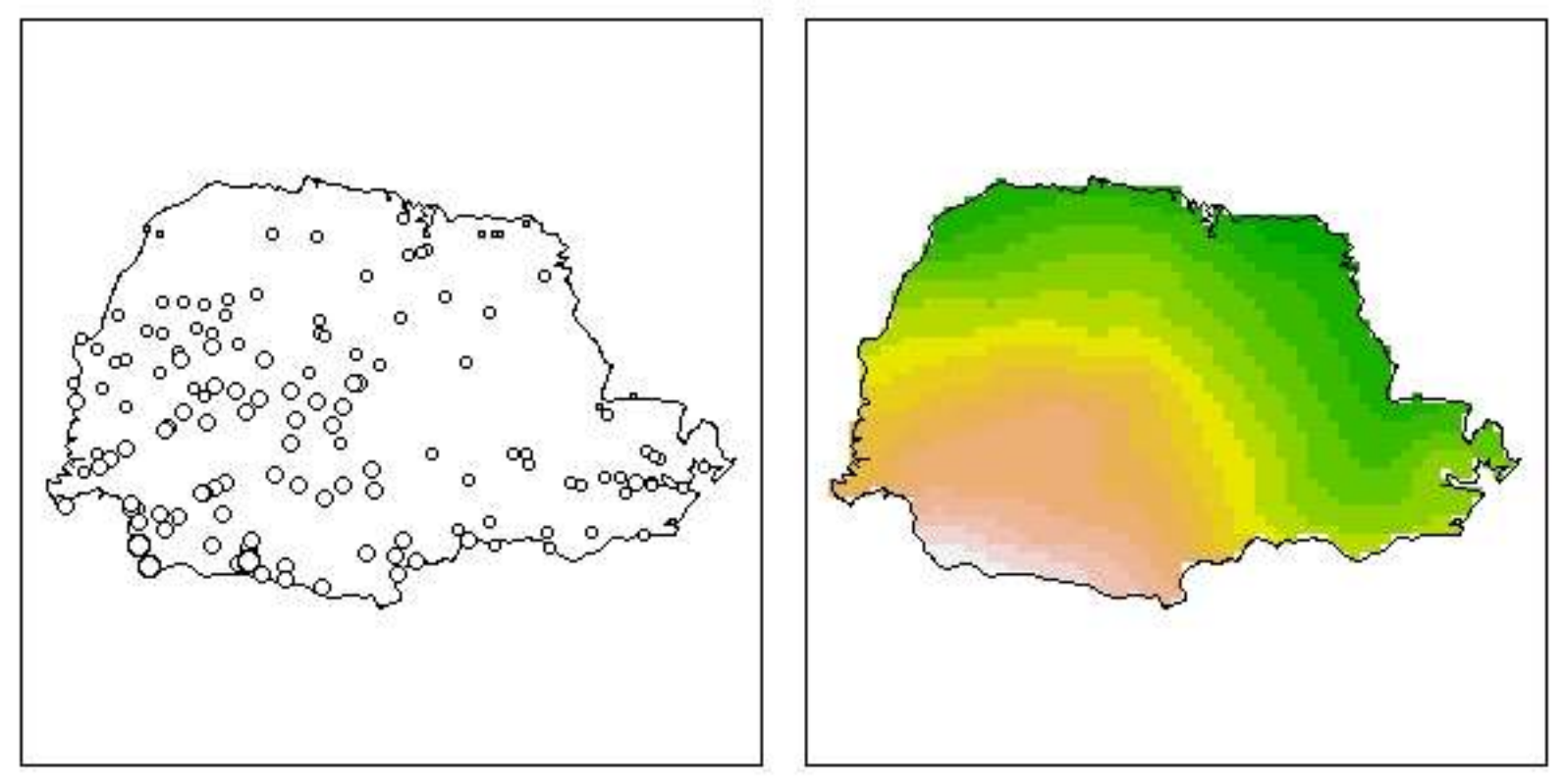
- Rule for aRT scripts: all data using database



aRT script

```
conn       = openConn(user="pedro")
db         = openDb(conn, "dbname")
thpoints   = openTheme(db, "themepoints")
thborder   = openTheme(db, "themeborder")

points = getPoints(thpoints)
border = getPolygons(thborder)
data   = getData(thpoints)

raster = krige(points, border, data)

l = createLayer(db, "lraster")
addRaster(l, raster)
thraster = createTheme(l, "themeraster")
```

aRT results



How to integrate

- Option 1

Execute R from the GIS and run the entire script

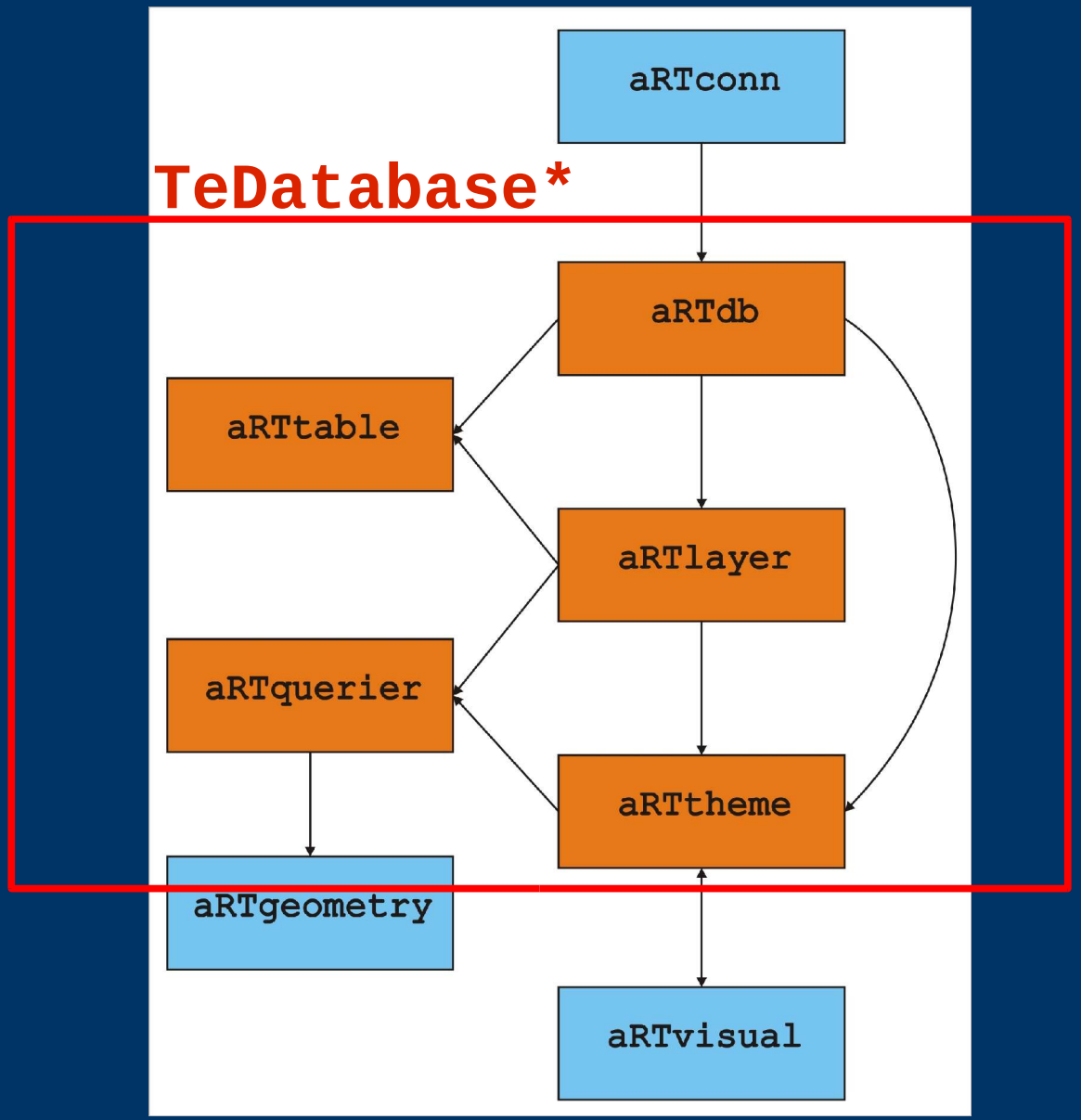
Problem: two connections to the database, and after that the GIS must reconnect to read metadata

- Option 2

Use aRT internal functions to read from the database and build R objects to be used

Problem: TerraLib programmers must know R and aRT internally.

How to integrate: Option 3



- Share a database connection (TeDatabase)
- Require little changes in the original script

Example of an aRT Script

```
conn       = openConn(user="pedro")
db         = openDb(conn, "dbname")
thpoints   = openTheme(db, "themepoints")
thborder   = openTheme(db, "themeborder")

points = getPoints(thpoints)
border = getPolygons(thborder)
data   = getData(thpoints)

raster = krige(points, border, data)

l = createLayer(db, "lraster")
addRaster(l, raster)
thraster = createTheme(l, "themeraster")
```

Wrapping an R code

```
tlKriege = function(db, tpointsname,
    tcontourname, lrastername, trastername){
  thpoints  = openTheme(db, tpointsname)
  thborder  = openTheme(db, tbordername)

  points = getPoints(thpoints)
  border = getPolygons(thborder)
  data   = getData(thpoints)

  raster = krige(points, border, data)

  l = createLayer(db, lrastername)
  addRaster(l,raster)
  thraster = createTheme(l, trastername)
  return(invisible())
}
```

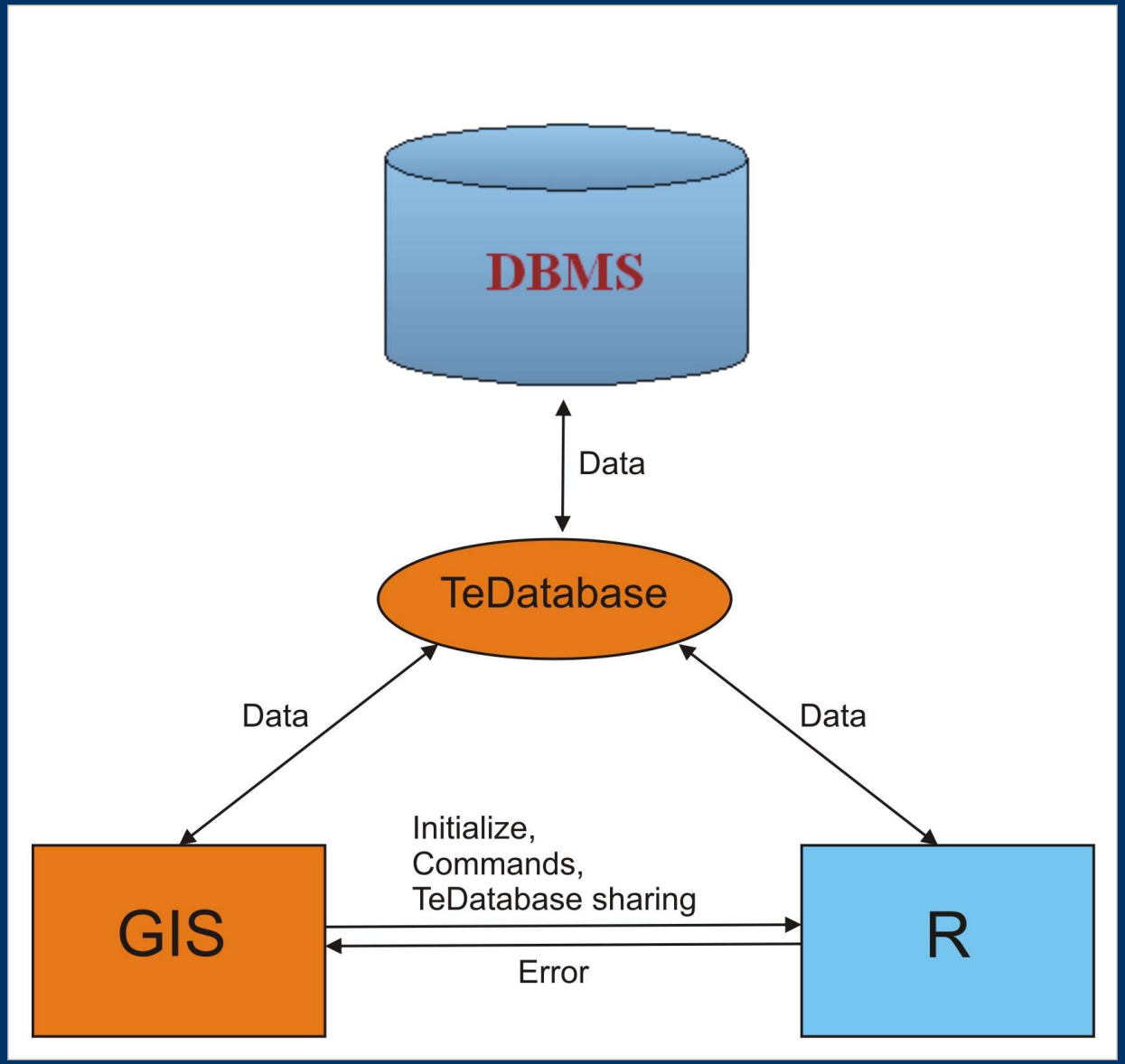
TeRwrapper

- Uses MyR library
- Operations:
  - Initialize the environment
  - Declare a TeDatabase
  - Execute an R command

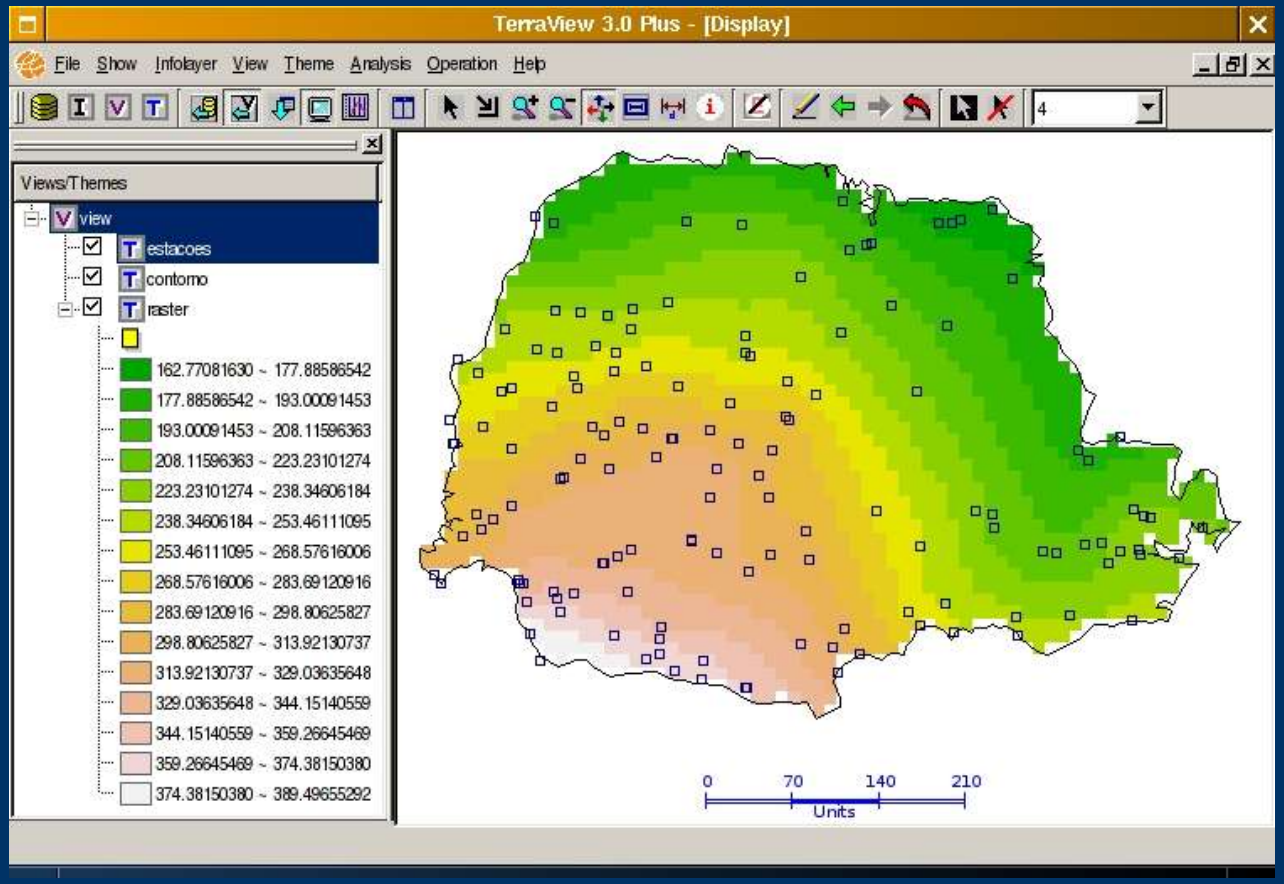
TerraLib code

```
TeDatabase* db;        char* str;
TeRwrapper* Rwrapper; string error;
...
Rwrapper = new TeRwrapper();
Rwrapper->declareDatabase(db, "db");
asprintf(&str, "tlKrige(db,%s,%s,%s,%s)",
          themepoints.c_str(),
          themeborder.c_str(),
          layerraster.c_str(),
          themeraster.c_str());
error = Rwrapper->execute(str);
if(error != "") ...

OnPaint();
```

Integration Diagram



Results



Note that aRT alone could do that, but the focus now is how to use this functionality without any R knowledge.

Conclusions

- Powerful analysis environment for statistics professionals
- Follows TerraLib project purposes
- Enables each agent to focus on his/her environment and skills

Future Work

- TerraCitrus, developing friendly interfaces
- Define a language for documentation of the functions, and try to generate part of the graphic interface for the programmers
- Ideas??

A Process and Environment for Embedding The R Software into TerraLib

Pedro Ribeiro de Andrade Neto
Paulo Justiniano Ribeiro Junior

GeoInfo 2005