Framework Domain

- Domain: Event-based data for visual display with countable keywords and optional priority
- Example data plugins:
  - Facebook posts with location/emoji
  - Webscripting a site such as TicketMaster with concert and song listings
  - Basic CSV file
- Example visualization plugins:
  - Frequency display
  - Roadmap of events with optimal routes


United States of America
State Boundary
Road
River
National Capital
State Capital
City or Town
0 300 KM
0 300 Miles
© 2007 Geology.com
CANADA
U.S.A.
MEXICO
Pacific Ocean
Atlantic Ocean
Gulf of Mexico
ALBERTA
BRITISH COLUMBIA
SASKATCHEWAN
MANITOBA
ONTARIO
QUEBEC
NEW BRUNSWICK
WASHINGTON
OREGON
IDAHO
MONTANA
WYOMING
NORTH DAKOTA
SOUTH DAKOTA
NEBRASKA
KANSAS
COLORADO
UTAH
NEVADA
CALIFORNIA
ARIZONA
NEW MEXICO
TEXAS
OKLAHOMA
MINNESOTA
IOWA
MISSOURI
ARKANSAS
LOUISIANA
WISCONSIN
ILLINOIS
MICHIGAN
INDIANA
OHIO
KENTUCKY
TENNESSEE
MISSISSIPPI
ALABAMA
GEORGIA
FLORIDA
SOUTH CAROLINA
NORTH CAROLINA
VIRGINIA
WEST VIRGINIA
PENNSYLVANIA
NEW YORK
MAINE
VERMONT
NEW HAMPSHIRE
MASSACHUSETTS
RHODE ISLAND
CONNECTICUT
NEW JERSEY
DELAWARE
MARYLAND
Lake Winnipeg
Lake Superior
Lake Michigan
Lake Huron
Lake Erie
Lake Ontario
Ottawa
Washington, D.C.
Nassau
THE BAHAMAS
Vancouver
Calgary
Saskatoon
Regina
Winnipeg
Moosonee
Victoria
Seattle
Olympia
Spokane
Portland
Salem
Boise
Helena
Bismarck
Pierre
Salt Lake City
Carson City
Sacramento
San Francisco
Los Angeles
San Diego
Phoenix
Denver
Cheyenne
Santa Fe
Lincoln
Topeka
Oklahoma City
Austin
Dallas
Houston
San Antonio
St. Paul
Minneapolis
Des Moines
Jefferson City
Little Rock
Baton Rouge
New Orleans
Jackson
Montgomery
Atlanta
Tallahassee
Nashville
Madison
Springfield
Chicago
Indianapolis
Lansing
Detroit
Columbus
Frankfort
Charleston
Richmond
Raleigh
Columbia
Harrisburg
Trenton
Dover
Annapolis
Albany
Hartford
Providence
Boston
Concord
Montpelier
Augusta
Toronto
Montreal
Quebec
Fredericton
Miami
Tampa
Jacksonville
RUSSIA
ALASKA
Anchorage
Juneau
HAWAII
Honolulu
Hermosillo
Chihuahua
Monterrey
Saltillo

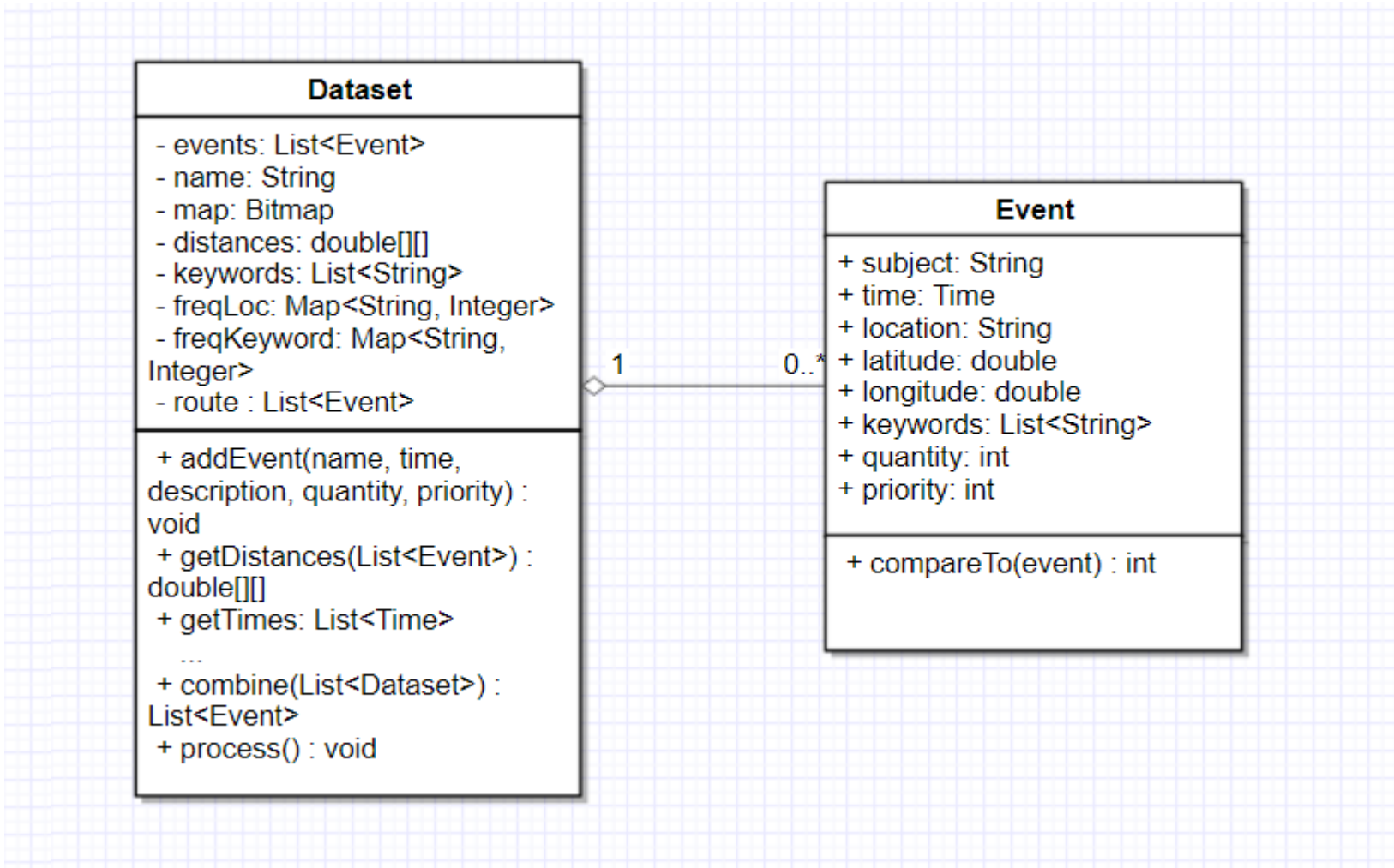
Framework Details

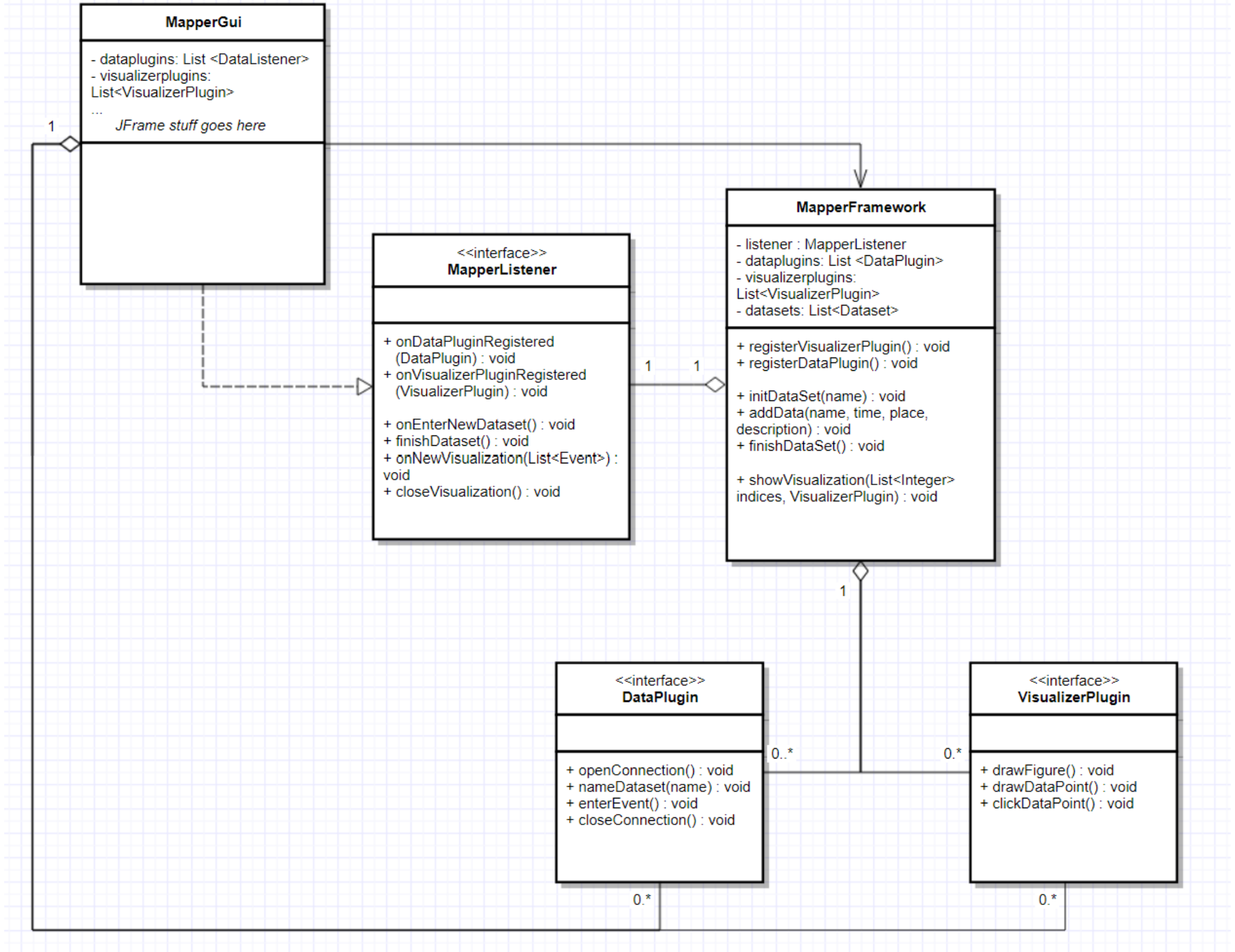
- This framework will be made to be reasonably general to handle most event based data in a particular region.
- Key abstraction:
  - Any event will have a time, place, and some description keywords.
  - Visualizing events on a map is the most natural presentation and is flexible.
- There is tons of flexibility in plugins:
  - Data would only require the bare minimum: time, place, and some description at minimum.
  - For visualizing, there are many modes of representing data on maps (e.g. frequency, path scheduling).

Project structure

- framework
  - core
    (I) MapperListener
    (C) MapperFramework
    (C) MapperGui
    (I) DataPlugin
    (I) VisualPlugin
    - datapoint
      (C) DataSet
      (C) Event

- plugin
  - data
    (C) TicketMasterPlugin
    (C) SpreadsheetPlugin
  - visual
    (C) MapRoutePlugin
    (C) KeywordFrequencyPlugin


Dataset
- events: List<Event>
- name: String
- map: Bitmap
- distances: double[][]
- keywords: List<String>
- freqLoc: Map<String, Integer>
- freqKeyword: Map<String, Integer>
- route : List<Event>
+ addEvent(name, time, description, quantity, priority) : void
+ getDistances(List<Event>) : double[][]
+ getTimes: List<Time>
...
+ combine(List<Dataset>) : List<Event>
+ process() : void
1
0..*
Event
+ subject: String
+ time: Time
+ location: String
+ latitude: double
+ longitude: double
+ keywords: List<String>
+ quantity: int
+ priority: int
+ compareTo(event) : int


MapperGui
- dataplugins: List <DataListener>
- visualizerplugins:
List<VisualizerPlugin>
...
JFrame stuff goes here
1
<<interface>>
MapperListener
+ onDataPluginRegistered
(DataPlugin) : void
+ onVisualizerPluginRegistered
(VisualizerPlugin) : void
+ onEnterNewDataset() : void
+ finishDataset() : void
+ onNewVisualization(List<Event>) :
void
+ closeVisualization() : void
1
1
MapperFramework
- listener : MapperListener
- dataplugins: List <DataPlugin>
- visualizerplugins:
List<VisualizerPlugin>
- datasets: List<Dataset>
+ registerVisualizerPlugin() : void
+ registerDataPlugin() : void
+ initDataSet(name) : void
+ addData(name, time, place,
description) : void
+ finishDataSet() : void
+ showVisualization(List<Integer>
indices, VisualizerPlugin) : void
1
<<interface>>
DataPlugin
+ openConnection() : void
+ nameDataset(name) : void
+ enterEvent() : void
+ closeConnection() : void
0..*
0.*
<<interface>>
VisualizerPlugin
+ drawFigure() : void
+ drawDataPoint() : void
+ clickDataPoint() : void
0.*
0.*