Serdica J. Computing **6** (2012), 89–100

Serdica
Journal of Computing

Bulgarian Academy of Sciences
Institute of Mathematics and Informatics

ASTROWEB – A GRAPHICAL REPRESENTATION OF AN ASTRONOMICAL WFPDB DATABASE USING A WEB-BASED OPEN SOURCE GEOGRAPHICAL INFORMATION SYSTEM[*]

Alexander Kolev, Milcho Tsvetkov, Dimo Dimov, Damyan Kalaglarsky

Abstract. The purpose of this study is to present the open-source MapServer and the way it can be used as a Web-based Geographic Information System. The MapServer object model is analyzed and algorithms are proposed for geo-referenced spatial data dynamic representation. The results of this analysis are used in the development of our application called AstroWeb. AstroWeb presents graphically the application data of the Plate Archives catalog (CWFPA v.2011) of Bulgarian Wide-Field Plate Database and ensures fast and easy access to the digitalized plate information.

1. Software tools for improving WFPDB user interface. The Bulgarian Wide-Field Plate Database (WFPDB)[1] [3, 5] is a well-known and useful

ACM Computing Classification System (1998): H.5.2, H.2.8, J.2, H.5.3.

Key words: Geographic Information System (GIS), Web-based GIS, Graphical data web-access.

[*]This work is supported with Grant No. DO-02-275/2008 of the National Science Fund of the Bulgarian Ministry of Education, Youth and Science to the Institute of Information and Communication Technologies at the Bulgarian Academy of Sciences (IICT-BAS).

[1]`http://wfpdb.org/`

tool for professional astronomers with its previews of plate catalogues of interesting sky objects. The main questions in the process of using a well-working and table-based web-user interface are: (i) is it possible to design a more intuitive user interface and give amateur astronomers better chances to use WFPDB; and (ii) is there suitable software tools with which the graphical based web-interface of the astronomical WFPDB can be developed and improved. To our best knowledge, there is no available decision of web-based graphical representation for the large heritage of wide-field astronomical plates worldwide and for this reason we propose the use of a web-based Geographical Information System (GIS) as a possible solution.

One of the proven software applications available as open source and with possibilities for creating a web-based GIS is the product of the University of Minnesota called MapServer [6]. Originally designed for the needs of the National Astronautics and Space Administration (NASA) of the USA, the product was developed from the mid-90s of the last century, and its latest known performance was in July 2010. Over the years, the product has evolved its software architecture towards creating diverse and complete interfaces for embedding in various scripting languages.

The purpose of this research is to build up a web-oriented graphical user interface to astronomical WFPDB. The server-side application modules are designed via implementation of MapServer, known recently as PHP/MapScript [4]. The client side is supported only by JavaScript and DHTML and no third party plug-ins are necessary. The PHP/Mapscript server side module has free-of-charge license permissions, i.e., it can be applied without any restrictions, including the objectives of the discussed research.

In the paper we analyze the results of the use of PHP/Mapscript as an open-source GIS tool for graphical data representation within the WFPDB and for two algorithms to access WFPDB as well. The first algorithm is based on a *shape* file update approach, whereas the second uses creation of a vector layer during the working cycle.

2. *MapScript* object model. The Object model of the PHP/MapScript module is presented in Fig. 1. The main class in terms of object-oriented programming herein is the class Map that together with the rest of the hierarchically represented classes provides full operation ability of a web-based GIS. The most important classes directly related to the visualization of geographic and geo-referenced data are: Layer, Class, Style, Symbols.

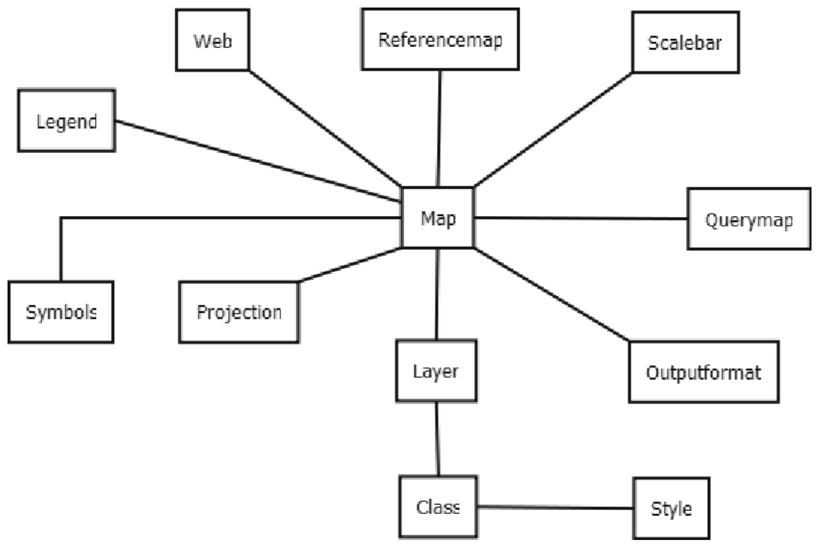


Fig. 1. The MapScript object model

The purpose of some classes shown in Fig. 1 is as follows:

- *Map*—principal class that controls input GIS data and the preliminary preparation by the server, and prepares the appropriate output file format for display purposes in the client's browser;
- *Layer*—defines the source of geographical and user data and ensures their showing. Geographical and user geo-referenced data can be in a standardized vector or raster format. The available geometric primitives in vector format can be point, line, multiline, polygon and multipolygon, which are widely used as vector graphic objects in GIS projects [1];
- *Class*—a named object containing the parameters of the type *style*, required for screen display. It has applications in the graphical representation of the results of geo-spatial queries too;
- *Style*—defines the type of parameters of graphical display: line type, form fill, color fill, outline color, line thickness, etc.;
- *Symbol*—used mainly for representation of geo-referenced objects defined by special characters. These characters can be given in a vector format, or raster format in a special bit mask, or as conventional characters of a given TrueType font.

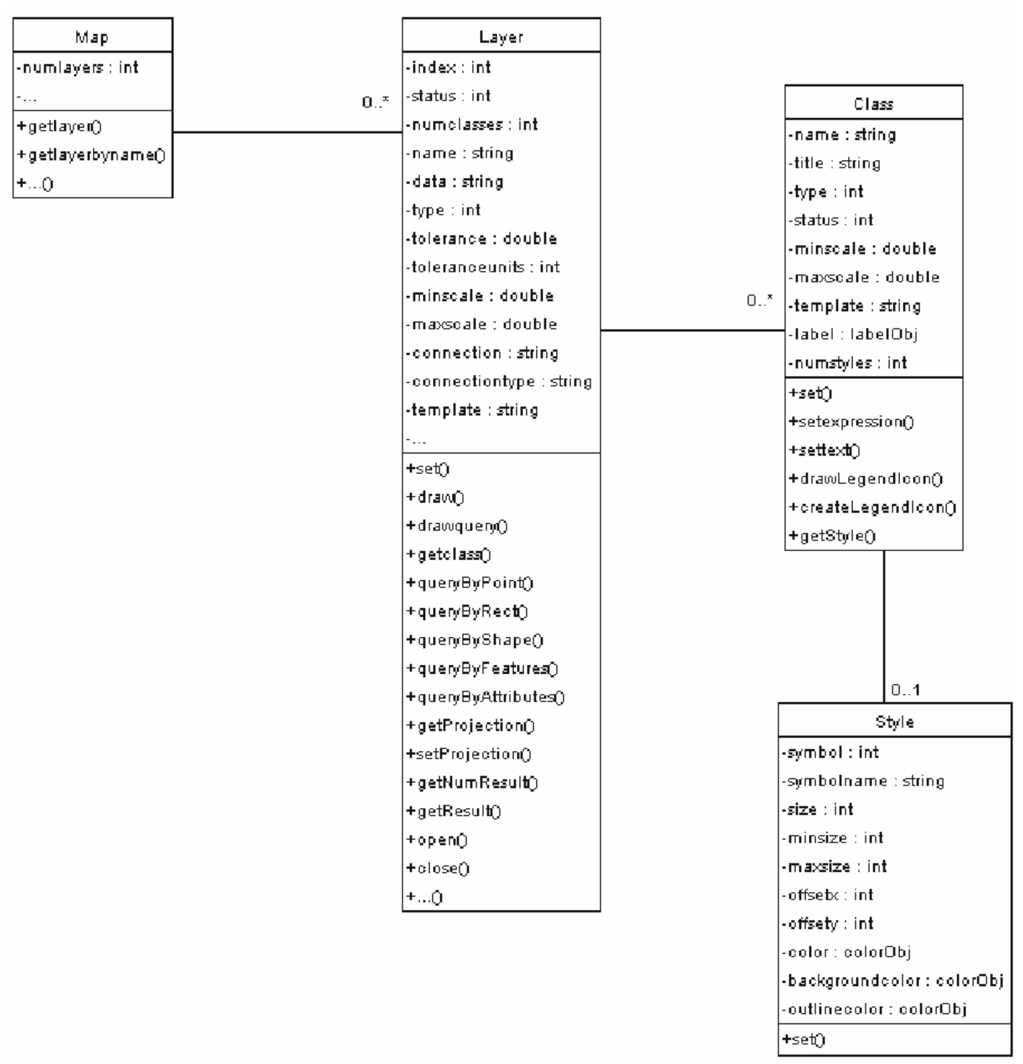


Fig. 2. The Layer class and its related classes

The *Layer* class together with its links to related classes is represented graphically in Fig. 2.

The classic software architecture of PHP/MapScript allows appliance through its configuration file (in a special text format). The configuration file given below initializes the values of the main features in the abovementioned classes. MapScript configuration files have the extension *map* by default.

An example of the .map file configuration content is as follows:

```
MAP                                        # beginning of the mapping configuration
  IMAGETYPE     PNG                        # specifies the type of graphics
  EXTENT        18.31 40.32 29.83 47.79    # determine the initial boundaries of
                                           # the Image in geographical coordinates
  SIZE          400 300                    # determine the boundaries of the
                                           # generated graphics in screen cords.
  SHAPEPATH    "/opt/fgs/astroweb/shp/"    # specifies vector data file path
  LAYER                                    # beginning of the vector layer def.
    NAME         skymap                    # vector layer name
    DATA         skymap                    # shape-file name
    STATUS       ON                        # determines the activity of the layer
    TYPE         POLYGON                   # determines the type of the layer
    CLASS                                  # beginning of the class definition
      NAME       "ispreview"               # class name
      STYLE                                # beginning of the style definition
        COLOR          240 100 100         # fill polygon color
        OUTLINECOLOR     0 255 255         # outline polygon color
      END                                  # end of the style definition
    END                                    # end of the class definition
  END                                      # end of the layer definition
END                                        # end of the mapping configuration
```

The software configuration shown above is a working example of a graphical visualization of an electronic map. To ensure the necessary GIS features for our system, i.e. to deliver large and selectable amounts of WFPDB data to a wide spectrum of users, we perform a detailed examination of information interactions during the programming cycle that is given below.

3. Graphical representation of WFPDB data.

A possible approach for the representation of WFPDB data in the software environment of PHP/MapScript is the rewriting and reload of the *shape* vector graphics file during the program execution cycle, see Fig. 3. The example below shows the shape file in the ESRI format [2] named *Sky Map*. The ESRI shape file is a widely used file format in commercial and open-source GIS projects for representation of vector-based geo-referenced features.

Let us consider the programming cycle represented in Fig. 3. The information flow can be described as follows:

- Creation of an object *objMap*;
- Loading of the static configuration file *config.map*;
- *Update map* procedure running a script that reads the WFPDB data and generates a custom *shape* file to be stored in the file system and to update the respective configuration data in the system *config.map* file;

- The object *objMap* performs a *draw* method (without parameters) and creates an object *objImage*, which contains the desired graphic;
- The client browser is instructed to expect a data flow of *png image* type;
- The object *objImage* performs a *saveImage* method (without parameters), which sends a graphical data flow for visualization in the client browser;
- The program cycle is closed with an expectation for user's intervention—*user interaction event wait.*

Another possible approach for presenting WFPDB data uses the possibility of creating a new vector layer, not described in the configuration file, as shown in Fig.4. We will discuss only the differences in information flow compared to the previously described approach, see Fig. 3:

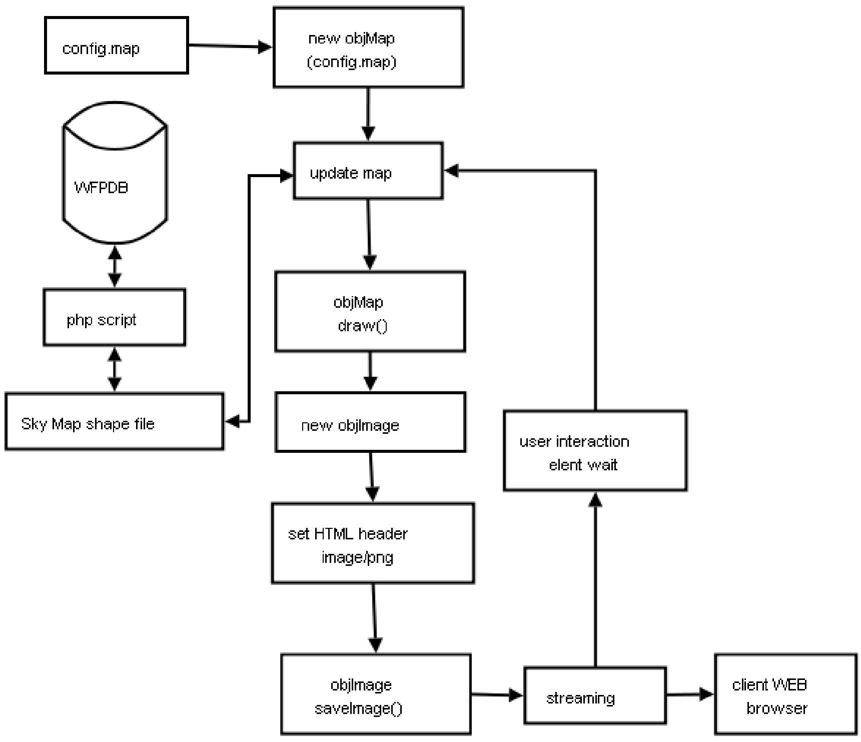


Fig. 3. Representation of the shape file update algorithm

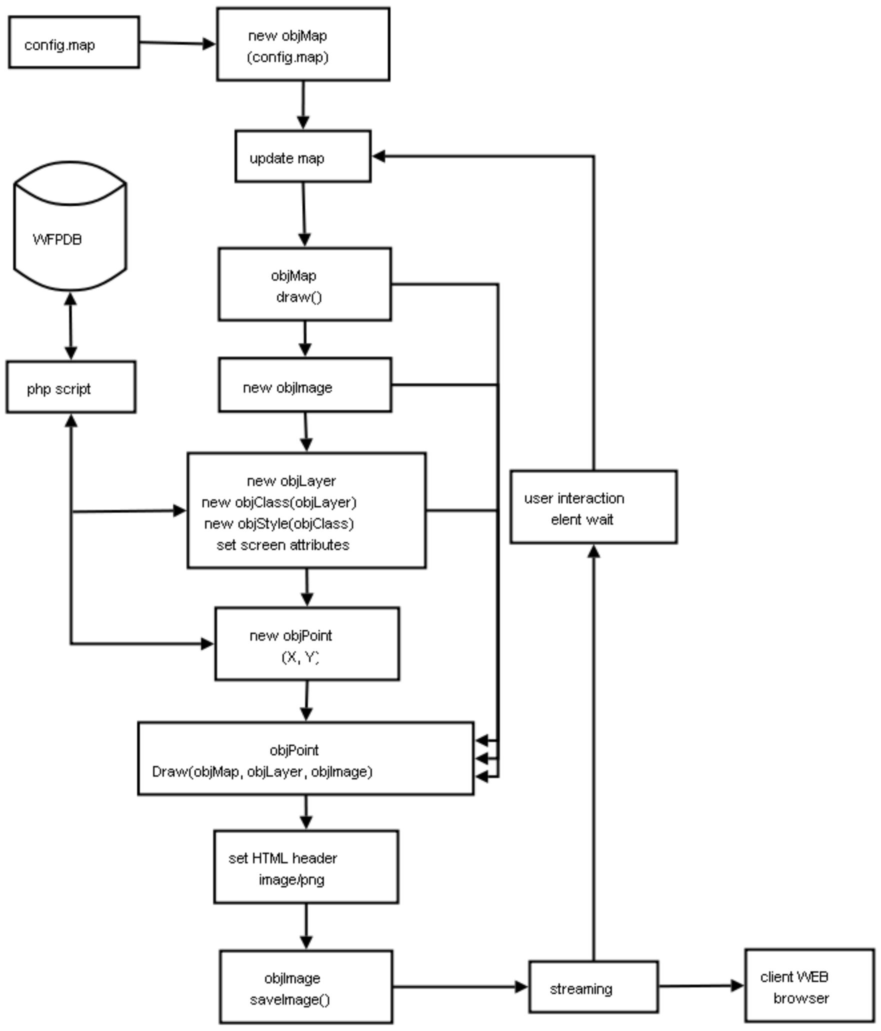


Fig. 4. Representation algorithm with creation of a vector layer

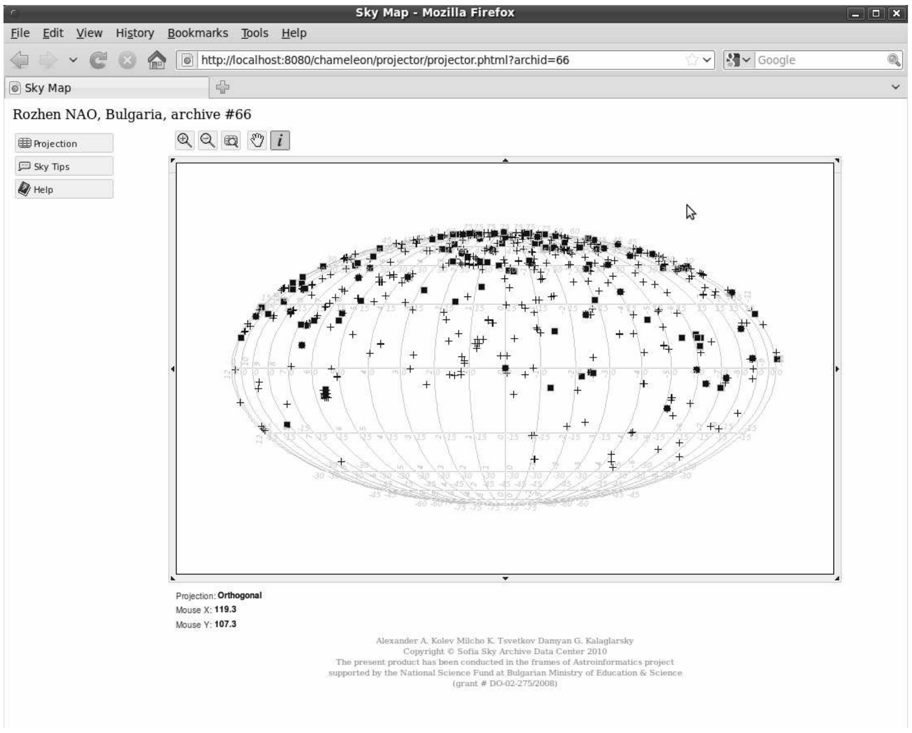
- Creation of the objects *objLayer, objClass, objStyle*;
- Exhaustive reading of WFPDB and setting up the properties that manage the on-screen representation of dynamic user information, such as color, scale, character, etc.;
- Creation of those *objPoint* geometric objects that use transformed coordinates from WFPDB;

- For each new geometric object created the *draw* method is executed with the parameters *objMap, objLayer*, and *objImage.*

The rest of the information stream herein is similar to the shape file update algorithm previously described.

4. Results. The algorithms and data structures discussed above were tested for the needs of the Astroinformatics project [7]. A brief description of the software solution follows hereinafter.

Starting an interaction with WFPDB via the web-interface panel as illustrated in Fig. 5, the user can choose the astronomical observatory of his/her interest and select among the astronomical images available therein. To this end the server PHP script reads the custom data from WFPDB and generates the SkyMap Shape file that is displayed in the graphic window of the panel. For the time being only the graphical archives of the Rozhen National Astronomy

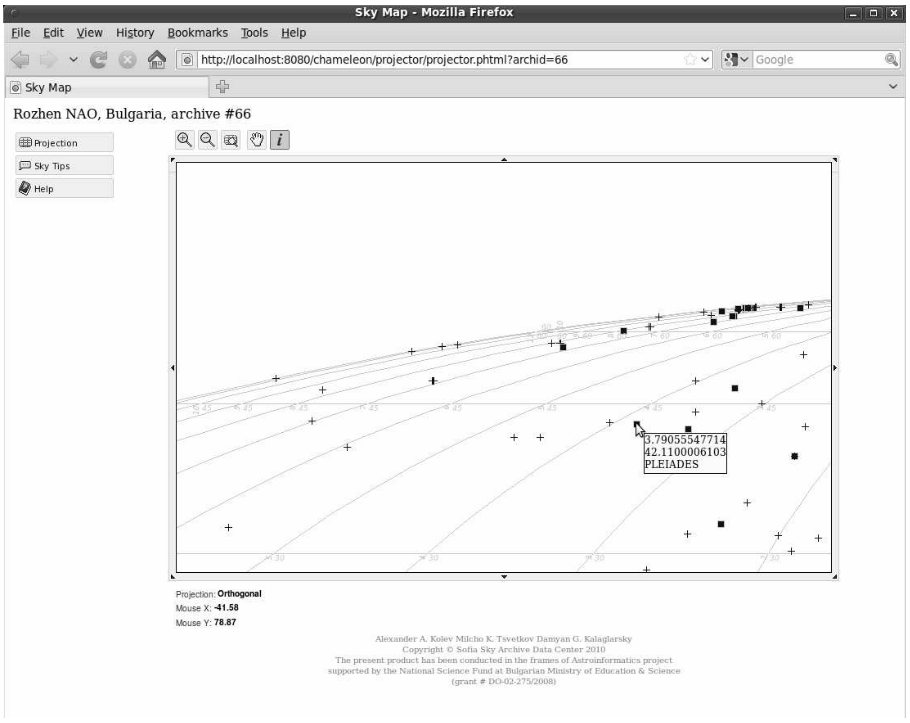

Sky Map - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8080/chameleon/projector/projector.phtml?archid=66
Google
Sky Map
Projection
Sky Tips
Help

Rozhen NAO, Bulgaria, archive #66

Projection: **Orthogonal**
Mouse X: **119.3**
Mouse Y: **107.3**

Alexander A. Kolev Milcho K. Tsvetkov Damyan G. Kalaglarsky
Copyright © Sofia Sky Archive Data Center 2010
The present product has been conducted in the frames of Astroinformatics project
supported by the National Science Fund at Bulgarian Ministry of Education & Science
(grant # DO-02-275/2008)

Fig. 5. Preview of SkyMap Shape file of the Rozhen NAO archive


Sky Map - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Google
Sky Map
Projection
Sky Tips
Help

http://localhost:8080/chameleon/projector/projector.phtml?archid=66

Rozhen NAO, Bulgaria, archive #66

3.79055547714
42.1100006103
PLEIADES

Projection: **Orthogonal**
Mouse X: **-41.58**
Mouse Y: **78.87**

Alexander A. Kolev Milcho K. Tsvetkov Damyan G. Kalaglarsky
Copyright © Sofia Sky Archive Data Center 2010
The present product has been conducted in the frames of Astroinformatics project
supported by the National Science Fund at Bulgarian Ministry of Education & Science
(grant # DO-02-275/2008)

Fig. 6. Zoomed Sky Map with a selected WFPDB object therein

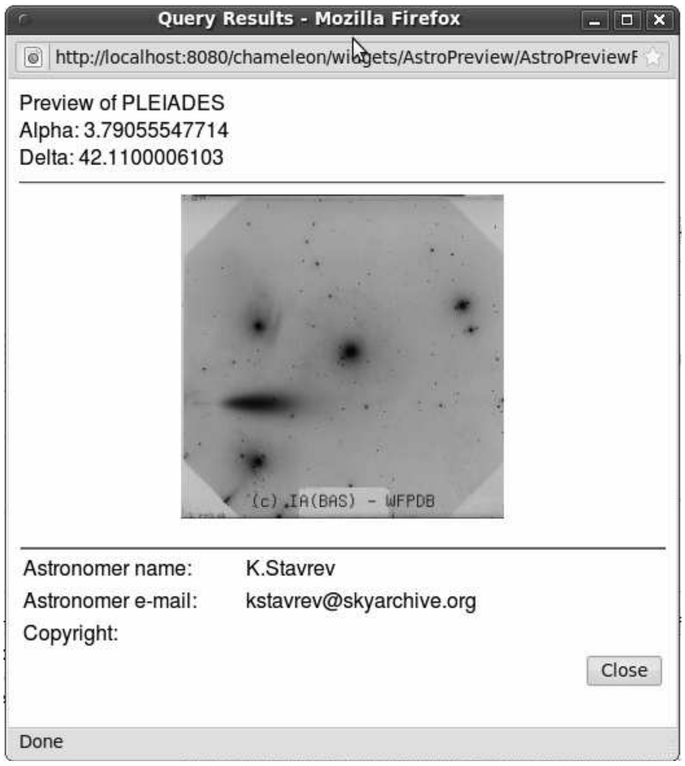
Observatory (NAO) can be retrieved from WFPDB.

In the next step the user can navigate simultaneously using the “zoom” and “pan” tools of the interface. He/she can also point with the mouse cursor to interesting astronomical objects, see Fig. 6. A special “info” tool is also provided by the interface. This tool activates customized screen hints that dynamically tell the user which WFPDB object is currently selected and what its features are.

And at the last step, a single click of the mouse on the selected astronomical object instructs the server script to retrieve a preview of the astronomical image from WFPDB and to present it in a separate web-browser window for detailed observation, as shown in Fig. 7.

5. Conclusions.

We designed a web-based graphical interface to the astronomical WFPDB in the frames of the experimental AstroWEB information system and proposed two algorithms for it. The first algorithm is a classical approach and is characteristic of the pure MapServer tool, whereas the second one


Query Results - Mozilla Firefox
http://localhost:8080/chameleon/widgets/AstroPreview/AstroPreviewF

Preview of PLEIADES
Alpha: 3.79055547714
Delta: 42.1100006103

Astronomer name:	K.Stavrev
Astronomer e-mail:	kstavrev@skyarchive.org
Copyright:	

Fig. 7. A preview of the selected WFPDB object

is based on an approach specific for the advanced PHP/Mapscript environment. Both algorithms are concurrently realized in our AstroWEB application.

Some new GIS features, tailored for the advanced user of astronomical databases, are also added here, such as: navigational "pan" and "zoom", dynamical info on the selected sky object, etc.

The web-based nature of the AstroWEB system provides clear distinctive conditions of operation either for the server subsystem, or for the system clients with their different user expectations for the information retrieval by the system. In general, AstroWEB is designed as a multi-user system and its users can access a dynamically generated graphical content, from different domains at the time of query, on the fly, and real time. In this context both presented approaches can be briefly discussed in terms of advantages and weaknesses, as follows:

- Advantages and disadvantages of the approach with a renovated shape file: (i) It does not require thorough knowledge of the object model *MapScript*,

(ii) The program module for creation of an updated shape file can be developed by a third-party, and (iii) The performance of a full version of AstroWEB doesn't need very large resource and time. As a technological shortcoming of this approach, the relatively intensive operation-load of the server file system can be pointed out, as well as the necessity of a lock/unlock subsystem for scheduling the various file operations that are specific for the operating system of a particular server.

- Advantages and disadvantages of the approach with dynamical creation of vector layer: (i) It does not overload the server file system and generally has greater productivity in terms of real system load. The following disadvantages can be found in the approach: it needs detailed knowledge about used object models and relatively high programming and testing skills for system development and/or upgrade.

REFERENCES

[1] ArcGis Resource Center, Geometry properties. `http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#//006z00000007000000`, February 2012.

[2] ESRI Shape File Technical Description, An ESRI White Paper – July 1998. `http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf`, March 2007.

[3] Kalaglarsky D. Wide-Field Plate Database – Sofia Search Page. `http://draco.skyarchive.org/search/`, September 2009

[4] PHP Mapscript. `http://maptools.org/php_mapscript/`, April 2008

[5] Tsvetkov M. K., K. Y. Stavrev, K. P. Tsvetkova, A. S. Mutafov, E. H. Semkov. Wide-Field Plate Database. Institute of Astronomy, Bulgarian Academy of Sciences, 1997.

[6] UNM MapServer. `http://mapserver.gis.umn.edu/`, April 2008

[7] Kounchev, O., M. Tsvetkov, D. Dimov, et al. Astroinformatics: A Synthesis between Astronomical Imaging and Information & Communication Technologies. Modern Trends in Mathematics and Physics, ISBN 978-954-580-264-5, *Bulg. J. Phys,* **32** (2009), No 2, 60–69.

Alexander Assenov Kolev
Defense Institute
34, Totleben Blvd.
1606 Sofia, Bulgaria
e-mail: `alexkolev@yahoo.com`

Dimo Dimov
Institute of Information
and Communication Technologies
Bulgarian Academy of Sciences
Acad. G. Bonchev St., Bl. 2
1113 Sofia, Bulgaria
e-mail: `dtdim@iinf.bas.bg`

Milcho Tsvetkov
Institute of Astronomy
and National Astronomical Observatory
72, Tsarigradsko Shose Blvd
1784 Sofia, Bulgaria
e-mail: `mtsvetkov@astro.bas.bg`

Damyan Kalaglarsky
Institute of Astronomy/SSADC
Bulgarian Academy of Sciences
72, Tsarigradsko Shose Blvd
1784 Sofia, Bulgaria
e-mail: `damyan@skyarchive.org`

Received October 31, 2011
Final Accepted February 20, 2012