PHP MapScript Applications at NACSE: Integrating Content from Research Databases

Dylan Keon

NACSE – Oregon State University

MapServer Users Meeting – June 2003

Background

- NACSE
    - Grant funded research group at Oregon State University with strengths in:
        - HPC
        - database-to-web application development
        - usability engineering
    - Funded by NSF, USFS, BLM, NBII, DoD, others
    - Maintain production databases such as:
        - Nationwide USFS Lichen/Air Quality Monitoring Program
        - Herbarium specimen collections from several universities
        - Medical Nuclide database for OSU Nuclear Engineering Dept.
        - International Transboundary Freshwater Dispute database
        - Catalogue of Oregon Marine and Coastal Information database

Background

- GIS at NACSE
  - Primarily produce web-based mapping interfaces (MapServer, ArcIMS)
  - Do some data processing and manipulation with ArcInfo (AMLs), GRASS, and Erdas Imagine, also use MapScript, GDAL, etc. to do some processing
  - Most GIS activities are in support of scientific (often biological or ecological) research databases
  - Moving toward making all database content available via both text query and mapping interfaces

MapServer or ArcIMS?

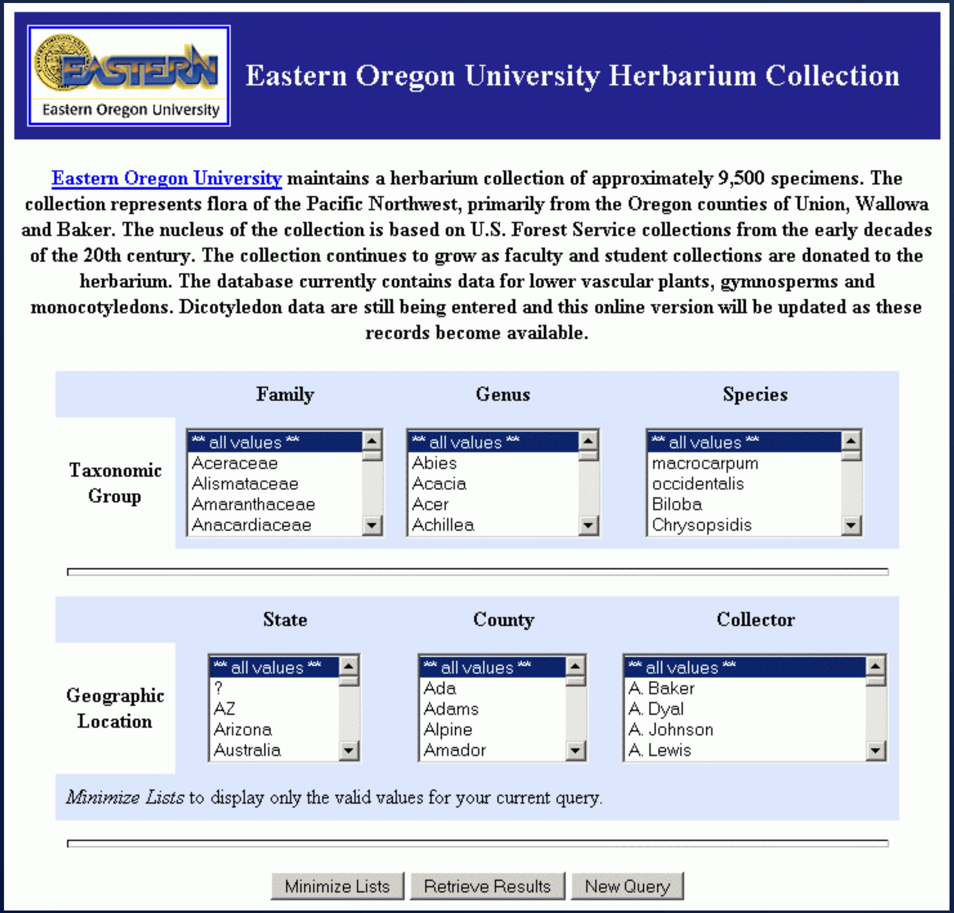
- **Started developing with ArcIMS in 2000**
  - **Pro: Nice selection of tools out of the box (buffering, etc.)**
  - **Pro: Site license = minimal fees for us**
  - **Con: Few data types can be used**
  - **Con: Heavy on resources, mediocre performance**
  - **Con: Database connectivity and other customization fairly complicated – really wanted to leverage Perl, PHP skills**
- **Moved most development to MapServer in 2002**
  - **Con: Fewer tools available, most need to be custom-built**
  - **Pro: Excellent performance**
  - **Pro: Accesses wide range of data types**
  - **Pro: MapScript flavors provide impressive power and flexibility, database connectivity fairly easy via PHP or Perl**
  - **Pro: Open Source Software ☺, easy to implement changes**
  - **Pro: Great user community and mailing list support**

QML → PHP → MapServer

- QML (Query Markup Language)
  - Perl CGI developed at NACSE
  - Drives many of our database-to-web interfaces
  - Tags-based language that can be interspersed with HTML
  - Fairly easy to learn

```
<td align=“center”><b>Watershed</b><br>

<qml_input name="MajorWatershedIn[ ]" sql="select distinct
MajorWatershed $siteGenus order by MajorWatershed" size="10"
multiple>

</td>

<td align=“center”><b>Group</b><br>

<qml_input name="ClassIn[ ]" sql="select distinct lower(Class) as
Class $siteGenus order by Class" size="10" multiple>

</td>
```

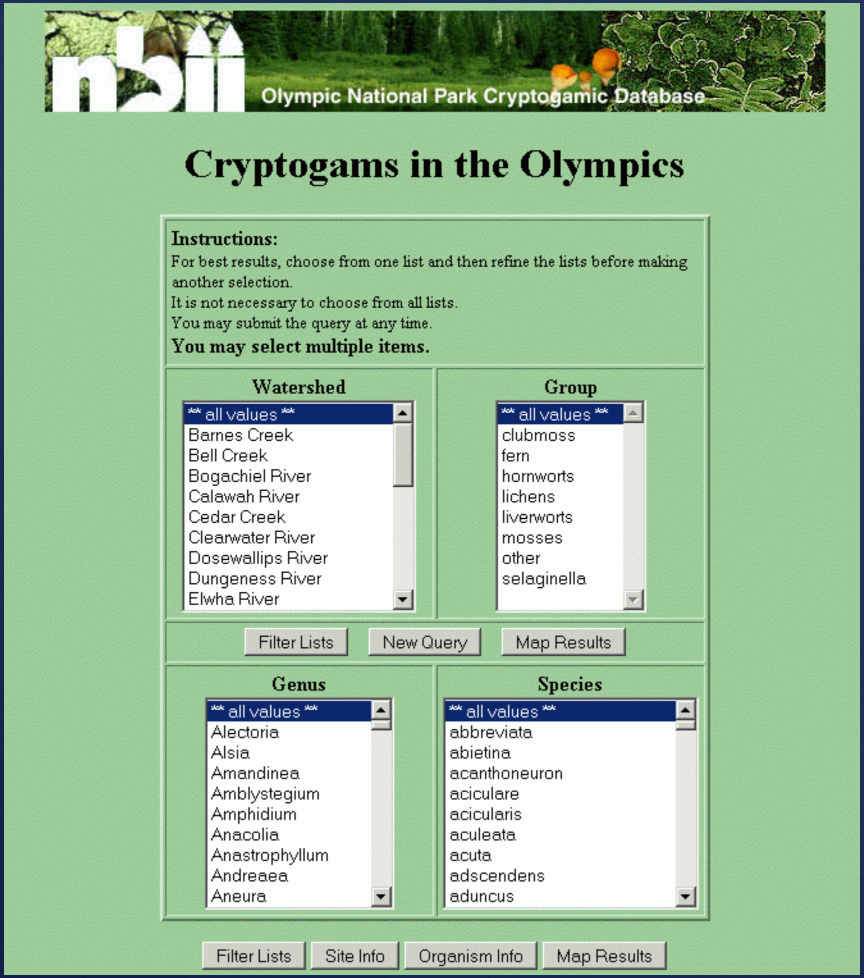
QML → PHP → MapServer



Eastern Oregon University Herbarium Collection

Eastern Oregon University maintains a herbarium collection of approximately 9,500 specimens. The collection represents flora of the Pacific Northwest, primarily from the Oregon counties of Union, Wallowa and Baker. The nucleus of the collection is based on U.S. Forest Service collections from the early decades of the 20th century. The collection continues to grow as faculty and student collections are donated to the herbarium. The database currently contains data for lower vascular plants, gymnosperms and monocotyledons. Dicotyledon data are still being entered and this online version will be updated as these records become available.

Minimize Lists to display only the valid values for your current query.

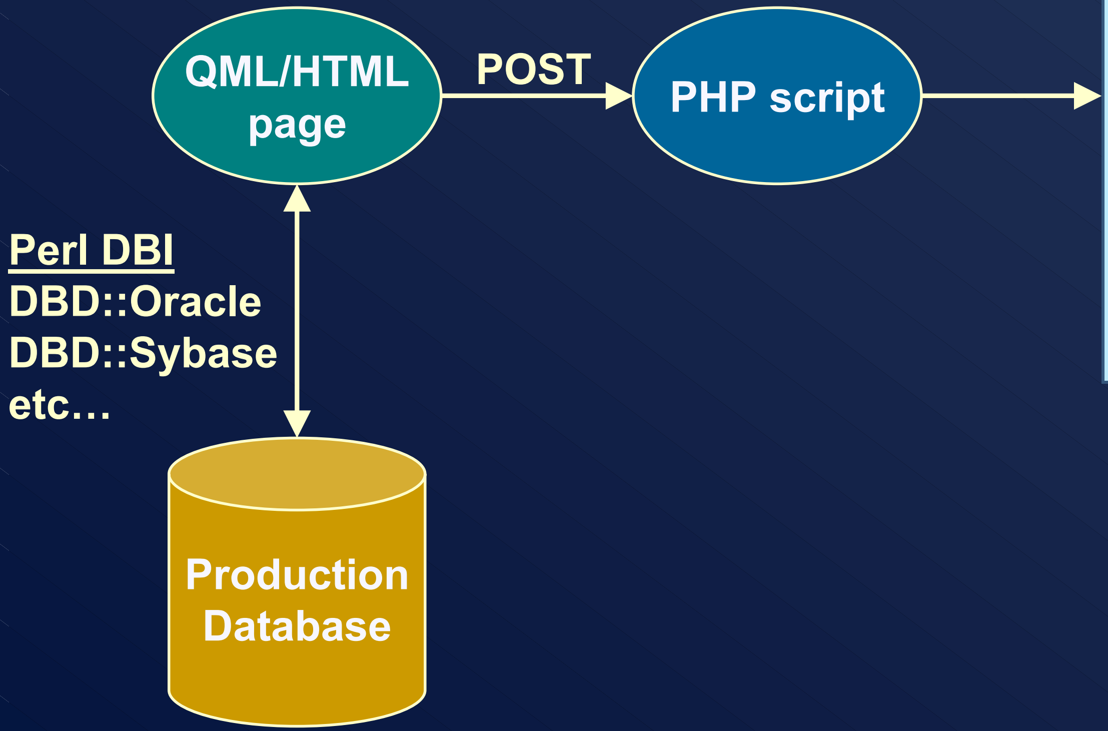


Olympic National Park Cryptogamic Database

Cryptogams in the Olympics

Instructions:
For best results, choose from one list and then refine the lists before making another selection.
It is not necessary to choose from all lists.
You may submit the query at any time.
You may select multiple items.

QML → PHP → MapServer



QML/HTML page

POST

PHP script

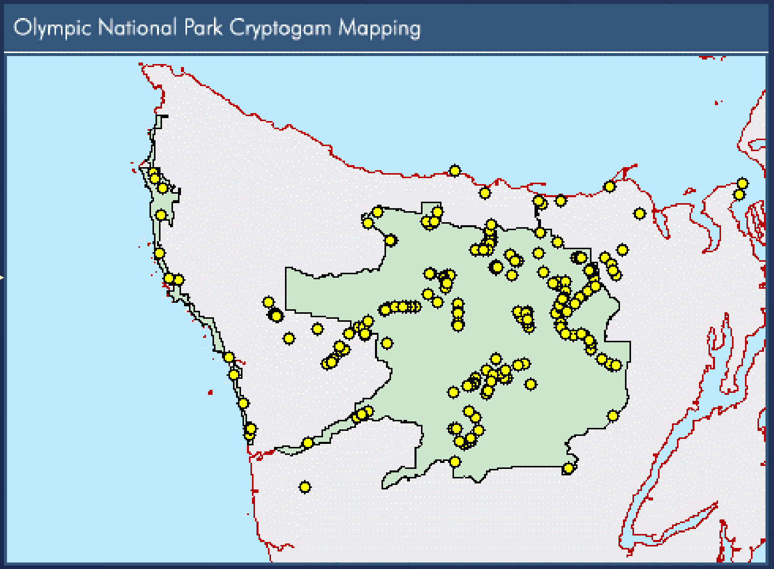
Perl DBI

DBD::Oracle

DBD::Sybase

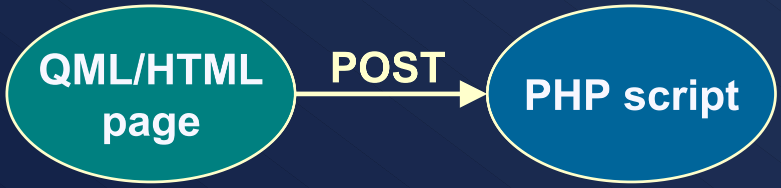
etc…

Production Database



Olympic National Park Cryptogam Mapping

QML → PHP → MapServer



- After data are passed from QML page to PHP script:
  1) Parse posted name[ ]=value pairs into arrays – PHP identifies each “name[ ]” as an array name when passed via GET or POST and automatically populates the arrays
  2) Dynamically formulate SQL query based on contents of the array(s)
  3) Send SQL query to database using normal PHP functions
  4) Generate shape features and dbf file by looping through the resultset
  5) Provide query results from cached shp/dbf files, which are tied to user’s session ID

QML → PHP → MapServer

- Dynamically creating a queryable point layer
  - Three possible ways to do this (more probably exist):
    1) Create shape features (and dbf files) from the incoming data, save to disk for use on map and to provide query results
    2) Create shape or layer features in memory from the incoming data, fire off new query to database every time a point is queried
    3) Create layer features in memory from the incoming data, store attributes in array with same index as associated point shapes, provide query results from that array
- We chose shp/dbf file creation because…
  - Easy to do queryByPoint(), etc. on the dummy layer, which looks just like a normal static shapefile to MapServer
  - Maintain user's state – the resultset is always available to be queried since shapefiles are on disk and tied to user's session ID
  - Performance is fine (shapes, dbf file created from a ~10,000 row resultset take just a few seconds to generate)
  - Might also try option #3 above, though…

Example – Cryptogams of ONP

- Cryptogams = mosses, lichens, liverworts, etc.
  - Abundance of these organisms in Olympic National Park – park scientists have amassed > 10,000 records from research sites
  - NACSE developed the relational database, QML-driven web interface to that database, and MapServer application to visualize the data
  - Work was funded through the National Biological Information Infrastructure (NBII)
  - Researchers and managers wanted to query research site data, interactively view the results in their spatial context, and retrieve site-specific data via the mapping interface
  - Template for further work

Example – Cryptogams of ONP

- Online demo:
  - http://gis.nacse.org/cgi-bin/qml/crypto.qml
    - (Visit http://gis.nacse.org for more information about this site, including instructions)

Example – Sevilleta LTER Site

- LTER = Long Term Ecological Research Program
  - Current interface supports Sevilleta LTER Site, which is located in New Mexico
  - Will be extended to all 24 LTER sites, user can quickly switch among them in a single interface
  - Site data from research plots can be integrated on-the-fly, as in the Cryptogams example
  - Dynamically displays remotely-created raster output from the Spatial Data Workbench (SDW)
    - User creates imagery analysis output, chooses to map it
    - Request is sent to NACSE
    - PHP page grabs the image and georeferencing info
    - Image is added to interface for correct LTER site, legend is dynamically updated

Example – Sevilleta LTER Site

- Online demo:
  - http://gis.nacse.org/lter/sev.php
    - (Visit http://gis.nacse.org for more information about this site, including instructions)

Additional PHP MapScript Applications

- AVIRIS flight line reference map generator
  - Many AVIRIS flights have been completed over LTER sites
  - LTER Network Office needed a way to create reference maps to display flight paths as links to available imagery
  - Takes user-submitted coordinate input, creates and draws line features across chosen LTER site
  - Supports both Geographic and UTM projections, offers several image sizes
- Nongeoreferenced imagery viewer
  - Scaled-down version of LTER interface
  - Allows users to view nongeoreferenced input imagery, while still using tools such as zoom, pan, etc.
  - Reads information from imagery header files

Future Plans

- Integrate rubber-band box DHTML tools for zoom/pan/query
- Move to PostgreSQL/PostGIS framework
- Enhance template to allow student workers to easily create mapping interfaces that link to their QML pages and databases

For More Information

- Dylan Keon

  keon@nacse.org

- GIS at NACSE

  http://gis.nacse.org

- QML

  http://www.nacse.org/qml