sci-wms CI-2014 Outline

Brandon A. Mayer

1 Possible Titles

1. sci-wms: A python-based web map service for visualizing geospatial data

2 Outline

1. U.S. IOOS Costal and Ocean Modeling (COMT) Testbed [**?**]

    1.1. *Vision*: Pg. 2 Paragraph 2

    1.2. "Increase Accuracy, reliability and scope of the federal suite of operational coastal and ocean modeling products to meet the needs of a diverse user community. Operational use covers a wide range of society-critical applications including forecasts, forensic studies, risk assessment, design and system management."

    1.3. An integral component of realizing the COMT vision are visualiztion tools that can enable rapid qualitative assesment of coastal maritime experiments potentially conducted by disparate entities, using different models, platforms and file formats. Sci-wms is a critical cyberinfastructure solution facilitating such a tool.

2. Web Map Service (WMS) [1]

    2.1. A standard developed by the Open Geospatial Consortium for delivering rasterized visual content in response to http requests.

    2.2. Standardizing the http interface separate users from datasets, allowing custom queries to be generated by users to which a Web Map Service can respond to by generating custom visual content.

3. sci-wms as a flexible Web Map Service (WMS)

    3.1. Sci-wms is a python based implementation of the OpenGIS Web Map Service [1]

    3.2. sci-wms implements the OpenGIS WMS standard [1] using the Django web application framework to handle http requests and responses and NumPy and Matplotlib [2, 3] for generating visual content.

3.3. Sci-wms is implemented using the django web framework with numpy [2] and matplotlib [3] as a backend for data manipulation and visualization.

4. Separating Topology and Variable Data

4.1. Costal forcasting and simulation models can contain many hundreds of Gigabytes of data, as such, an important design challenge for sci-wms is to replicate as little data as possible.

4.2. A conflicting operational requirement is that when content is requested by a client, sci-wms must respond in real-time to satisfy the request for visual content.

4.3. This is achieved by viewing a dataset as consisting of two separate entities, a topology and set of model variables.

4.4.

4.5. Therefore rather than reproducing model data locally, sciwms maintains a database of OpenDAP endpoints for accessing model data remotely.

4.6. Costal modeling experiments are conducted on a geo-registered grid. The type of which defines the model topology.

4.7. Example topologies are curvilinear, rectilinear, regular, and unstructured.

4.8. Define a view in the sci-wms context as a rendering of data at a particular geo-location.

4.9. When users request a view from sci-wms via the WMS http protocol sciwms

4.10. Sci-wms takes the unique approach of separating topology from variable data.

4.11. Costal modeling experiments are conducted on a geo-registered grid. The type of which defines the model topology.

4.12. Example topologies are curvilinear, rectilinear, regular, and unstructured.

4.13. Define a view in the sci-wms context as a rendering of data at a particular geo-location.

4.14. When users request a view from sci-wms via the WMS http protocol sciwms

5. NetCDF Markup Language (NCML)

5.1. Virtual Data layer

5.2. Provides CF-Compliant facad for raw model output.

5.3. Exposes each dataset, which may consist of numerous output files in a diverse format as a single CF-Compliant dataset accessable via OpenDAP as a netCDF data structure.

6. NetCDF CF-Metadata Conventions

6.1. Climate and Forcast (CF) - Metadata Convnetions

6.2. Designed to share files using the netCDF API

References

[1] Open Geospatial Consortium http://www.opengeospatial.org/standards/wms [Accessed: 2014-07-24], "OpenGIS Web Map Service."

[2] S. v. d. Walt, S. C. Colbert, and G. Varoquaux, "The numpy array: A structure for efficient numerical computation," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 22–30, 2011.

[3] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[4] pyugrid https://github.com/pyugrid/pyugrid Accessed: 2014-07-24.