

Reverse engineering of the GeoNetwork

Bin Ma u4446201

Supervisor:
Clive Boughton

1. Back Ground

GeoNetwork opensource is a standard based and decentralized spatial information management system. It is designed to access to Web Map Service databases from lots of data providers using meta data. When web map information is retrieved from other server, it can be shown as layers on the same map.

Intermap is a not well-documented component in GeoNetwork system. It become messy because of the large number of files, functions and codes without document. Further development and enhancement were hindered by these severe issues. So a well-documented component shall replace the Intermap.

OpenLayers is a JavaScript library independent from server for displaying map data in most web browsers. OpenLayers build rich web-based geographic applications using a Java Script API only at presentation layer. It can be plugged-in any other geographic application without concerns of which server and DBMS the application use. OpenLayers is a free software developed in the Open Source software community. It is a good choice to replace the Intermap.

2. Problem/Task description

Requirements:

1. All of the Javascript, HTML, XSL and XML code to do with the client side of GeoNetwork needs to be reverse engineered.
2. This will require determining all the Javascript functions, classes etc. and their description (in an easy to understand way) and their relationship with any other Javascript function.
3. Describe (in an easy to understand way) where in the XSL/XML/HTML code the Javascript functions are called.
4. Identify in which files the Javascript functions are to be found. Do a similar thing with the XML/XSL/HTML code.
5. Identify all unused functions.
6. Establish how InterMap might be able to be decoupled from GeoNetwork and an alternative is able to be "plugged in".
7. Investigate the potential use of the OpenLayers software for providing the "plugin" mentioned above.

Tasks:

1. A well structured XML is created to record all the functions in JavaScripts, include their name, description, subfunctions and link to each other.
2. A further linking matrix information will be established to record which XML/XSL/HTML file use the JavaScript functions.
3. Make the new components API implements the coupled API's functions to plug-out the old one and plug-in the new one.

Risks:

1. There are about 300 files and tens of thousands of lines. The project time could not be

enough.

2. There is a high complexity in the project involved with lots of (XML / XSL / JavaScript /AJAX) technologies. Understanding each technology and the inter-operation among them needs great effort.

Plan / Schedule

Week 1-4: Tasks: Initial preparation and plan

Learning Project Required Programming Languages

Milestone: Initial Presentation (12/08)

Project Plan (15/8)

Week 5-6: Tasks: Join Australian Geo-Science Presentation

Make JavaScript function records with XML document

Learning Project Required Programming Languages

Milestone: Put JavaScript functions into a XML file (29/8)

Week 7-8: Tasks: Mapping JavaScript functions to presentation layer sections.

Linking Matrix Establishment with XML document.

Milestone: Linking Matrix to index JavaScript functions.(05/9)

Function relationship map. (12/9)

Week 9-10: Tasks: Do statistics and indexing among functions and files.

Identify unused functions and similar functions.

Milestone: Unused functions and similar functions' report. (26/9)

Week 11: Tasks: Investigate the potential use of the OpenLayers software.

Give a evaluation report of the decoupling of the Intermap.

Milestone: Evaluation report of the decoupling of the Intermap (17/10)

Week 12: Tasks: Final report

Milestone: Final report (24/10)

Week 13: Tasks: Final presentation

Milestone: Final presentation (31/10)

References

O'Reilly Books:

Head First Serverlets and JSP

JavaScript The Definitive Guide

AJAX The Definitive Guide

XML in a nutshell