

Milestone 4 Report

uDig

March 31, 2005



Submitted To: Program Manager
GeoConnections
Victoria, BC, Canada

Submitted By: Jody Garnett
Refractions Research Inc.
Suite 400 – 1207 Douglas Street
Victoria, BC V8W 2E7
E-mail: jgarnett@refractions.net
Phone: (250) 383-3022
Fax: (250) 383-2140

TABLE OF CONTENTS

1	SUMMARY OF WORK ACCOMPLISHED	3
2	ENCOUNTERED PROBLEMS AND SOLUTIONS	4
2.1	CONTENT MANAGEMENT SYSTEM.....	4
2.2	WEB MAP SERVER STYLING SUPPORT.....	4
2.3	ECLIPSE FRAMEWORK API UPGRADES	5

1 SUMMARY OF WORK ACCOMPLISHED

The goal of this project is to build a GIS Desktop Client that integrates with Open GIS Consortium (OGC) Open Web Services (OWS).

The Application Framework we are using is called Eclipse Rich Client Platform. Eclipse Rich Client Platform is an offshoot of a popular Java Integrated Development Environment and offers a unique approach to Java User Interface development.

All required documents have been completed for this fourth and final milestone. These documents are:

Phase 4.10

- Final uDig User Guide
- Final uDig Developer's Guide
- Milestone Report (this document)

For Milestone 4 we have completed the following software release:

Phase 4.10

- uDig application version 1.0
- French language pack

We continue to be very pleased with the assistance provided by our partners and the open source community in supporting the uDig project.

This work is publicly available via the uDig project website, community wiki and Subversion repository.

2 ENCOUNTERED PROBLEMS AND SOLUTIONS

2.1 Content Management System

One of our goals is to market uDig as a Refractions Research innovation. To us this meant having a strong web presence where Refractions Research and uDig were synonymous. One of the steps we took towards this goal was to move uDig's Content Management System (CMS) to a new website whose URL resided within the Refractions domain (<http://udig.refractions.net>).

Originally we selected Drupal as our content management system of choice because Drupal:

- Is open source
- Has a large user base
- Was more flexible than its counterparts

Unfortunately, after a costly Drupal setup, we found that Drupal lacked significantly with regards to its ability to import content, link pages of content, and provide internationalization support. When evaluating the issues that were arising with our Drupal installation, we compared the functionality to the current CMS used, Confluence.

Eventually we acquired a copy of Confluence for the uDig project that could be hosted locally. Although this was not a perfect solution, it was significantly better than using Drupal. Confluence could both import and export content substantially better and allowed for easy scalable content linking and naming. There were a few drawbacks, for example it is more difficult to order content with Confluence. We also did not see any significant benefits from an internationalization standpoint.

2.1.1 Internationalization

Unfortunately internationalization was not supported in either CMS. Drupal's support was outdated and did not run, while Confluence did not have any support. In both cases the solution was the same – create a new CMS site. We have also created some re-direct macros for detecting and redirecting users based on their web browser's language. Although this is not a perfect solution because the content is not always synchronized, it is a good start.

2.2 Web Map Server

2.2.1 Query Support

uDig aims to treat all data in similar ways, independent of how the data is acquired. When it came time to gather information about an area on the screen,

we ran into some difficulties passing the request back to a Web Map Server (WMS).

To acquire information about a particular region in a WMS layer, the client issues a 'GetFeatureInfo' request. We started running into trouble when we found that the specification only supported pixels, which did not necessarily correspond to the screen. This resulted in uDig's functionality being changed to align closer with the constraints imposed by the WMS specification. We also applied reprojection algorithms to the screen data to form correct WMS requests.

The information acquired from the WMS was also problematic. This data was not standardized, ranging from Geographic Markup Language content to HTML content. We have leveraged the embedded web browser found within the Eclipse Rich Client Platform (RCP) to interpret and display the wide range of responses.

2.2.2 Styling Support

Styling arbitrary data is a lot simpler than it sounds due to the richness of the Style Layer Descriptor (SLD) specification; the only caveat being you need both the layer name, and the layer's geometry type.

When Styling a Web Map Server (WMS) layer, we already know the layer name, but unfortunately the layer's geometry type was not quite so easy to reverse-engineer. The WMS 'DescribeLayer' request allows the client to ask for a description of a particular layer. This description may optionally return a link to a Web Feature Server (WFS) from which you can determine the geometry type of the layer. Although this is all technologically possible, it took quite some time to complete our first request / response cycle.

Unfortunately though, many of the WMS instances did not return completed 'DescribeLayer' responses, leading to a small use case, and a large amount of yet incomplete effort. After completing more manual experiments we concluded that very few servers could accept an SLD document to style a 'getMap' request, and even fewer servers would adequately perform both a styled 'getMap' request and a 'DescribeLayer' request. At this time we have halted development of an extension to the WMS client, which would allow it to perform styled getMap requests as we felt the resources could be better spent on other aspects of the project. We look forward to re-opening the possibility of styled 'getMap' requests in the future when more WMS instances support this functionality.

2.3 Eclipse Framework API Upgrades

uDig leverages the Eclipse Rich Client Platform (RCP) to create a robust, extensible user experience. As the Eclipse project has published new Milestone releases we have evaluated the release, and in most cases have adopted the new Eclipse release. When Eclipse Milestone 5 and 5a (M5, M5a) were released we evaluated the new version. In this version there were some significant improvements to both the user experience and the build process for the Rich Client Platform. Unfortunately though, the Eclipse IDE was not up to the high standards we had come to expect of the Eclipse project. We have decided to wait

for the Eclipse Milestone 6 release prior to contemplating an Eclipse version change.

2.3.1 Wizards

Eclipse M5 showed a significant improvement for Wizard support. uDig was also undergoing many user-interface modifications. Ideally uDig would also make use of these improvements. In order to leverage the new Wizard API published in M5, we replicated the new M5 classes in the uDig codebase. This allows uDig to migrate to the Eclipse implementation on the next version bump of Eclipse within the uDig project.

2.3.2 Commands

As with many other applications, uDig has a notion of tools (Zoom, Info, etc.). When Eclipse M5 was released, they added a significant amount of core framework to help developers provide a rich, extensible tool set to their users.

The Workbench Command Framework (WCF) was design by Eclipse to provide:

- A Categorical Command Structure
- An Extensible Command Set
- Runtime Configurable and Persistable Bindings

Unfortunately this framework was core to the Eclipse M5 release and could not be easily broken out. This meant that we could not use the Eclipse solution directly, but rather only as inspiration for our Tool API.

Developers who are familiar with both the Command and Tool APIs will see a multitude of similarities. uDig does recognize the benefits of the WCF and intends to migrate to the WCF over time as the version of Eclipse leveraged by uDig is updated, and more extensive use of the Tool API develops. We recognize that this poses both a significant development risk for the code base in the future, as well as a maintenance risk at the present time.