

Web Report Editing Tool

Project Description

Client: Government Customer

Project Name: Web Report Editing Tool (WebRET)

The goal of this project was to create a single-page web application to allow for easy online editing of complex XML-based reports.

The Challenges

- ▶ Content was required to be read from a complex but standardized XML document format, the Intelligence Community Metadata Standard for Publications (IC-MSP), and saved back to IC-MSP format when editing was complete.
- ▶ The web interface needed to be easy to use and intuitive for users familiar with the many features of modern web apps.
- ▶ Power users needed quick, easy access to advanced functionality without interfering with less advanced users.
- ▶ Internet Explorer 8, a browser that does not support many web standards, had to be supported.
- ▶ WebRET needed to be testable to ensure quality control, to make sure it conformed to the existing specifications, and to provide a regression baseline for future development. As much as possible, the product needed to work as expected and prior functionality had to be tested to make sure it was not left out.
- ▶ The project needed to be completed within a short timeframe of only a few months.

The Solution

Agile-based Development : Because of the short window of time, agile methodologies were closely followed on WebRET. Daily standups of under 15 minutes, along with a weekly iteration planning meeting with the client to minimize miscommunication and keep the client in the loop, built a product that could be used immediately when it was handed off to the client.

Single-Page App Design : To convey the look and feel that the client desired, WebRET was designed as a single page web application, which allows for loading and saving reports without changing or refreshing the page. This was done by using primarily client-side JavaScript code for user interaction.

KnockoutJS : The client-side code in WebRET centered around KnockoutJS, an emerging JavaScript framework that follows the Model-View-ViewModel (MVVM) pattern. KnockoutJS separates the underlying data model, code and HTML view, creating an easily testable and maintainable framework for developing single-page web apps.

JRuby on Rails: For server-side operations, AIS used the JRuby on Rails platform. JRuby is an implementation of the Ruby language, with the additional key benefit of running on a Java Virtual Machine. This allows Ruby code to be hosted on existing Java application servers, such as Tomcat. JRuby also takes advantage of the benefits of Java, such as hardware independence and extensive garbage collection, while maintaining Ruby's rapid development style.

AIS AT - A - GLANCE

Founded

1982

Expertise

- Custom Application Development
- Tablet and Mobility Solutions
- Cloud Computing
- Infrastructure Engineering
- Operations & Support
- Strategic Consulting
- Software Engineering

Employees

400 +

Awards

- 2012 & 2011 Inc. 5000 Fastest Growing Companies
- 2010 Washington Business Journal's 50 Fastest Growing Companies
- 2008 Washington Business Journal's 100 Best Places to Work



APPLIED INFORMATION SCIENCES

2681 Commons Blvd.
Beavercreek, OH 45431
Phone: **937.999.2954**

www.appliedis.com

blog.appliedis.com

 twitter.com/appliedis

 [linkedin.com/appliedis](https://www.linkedin.com/company/appliedis)

 RSS

 [facebook.com/appliedis](https://www.facebook.com/appliedis)



Microsoft Partner

Gold! Portals and Collaboration
Gold! Software Development
Gold! Web Development

JRuby on Rails minimizes time spent setting up configuration by utilizing a convention-based architecture. It allows developers to skip much of the boilerplate code and start working directly on the underlying business problems. JRuby on Rails also fully supports RESTful actions, which made it ideal for WebRET's single-page web app format. With the extensive benefits of JRuby on Rails' speedy development process, extensive testability, built-in JSON endpoint support, and support for deployment on currently available infrastructure, it provided the tools needed for this time-limited project.

UX Design Patterns: WebRET was designed using modern patterns for user interface design. This included using patterns such as Edit-In-Place and Progressive Disclosure, as well as a focus on clean layout and usability. WebRET implemented Edit-In-Place across the report, allowing virtually all sections of the report to be edited with a simple click on the text. Likewise, the Progressive Disclosure pattern was used in the report's metadata, a group of additional fields that were infrequently edited. To make the metadata easily available but out of the way, AIS created a collapsible side panel to allow the user speedy yet unobtrusive access.

Testing Frameworks: AIS developed tests using three separate frameworks: Jasmine (for JavaScript unit tests), Selenium (for integration tests), and Cucumber (for Ruby testing and for tying all of the tests together). Jasmine tests were developed concurrently with all major data models and methods to ensure that written code was designed to be testable from the start. Each major feature was given Selenium integration tests before it was marked as completed.

Results

- ▶ A functional solution was delivered by AIS with only three months of development and one month of testing.
- ▶ WebRET has an extensive library of unit and integration tests available in the project, providing a regression system for safely adding future updates.
- ▶ The system was integrated into multiple client locations to provide a web-based standardized report editing system to many distinct user groups.