Field: Evolving a Rapid Prototyping Environment for Visually and Analytically Exploring Large-Scale Linked Open Data


ABSTRACT:

Field: meet the Web – 3D browsing

Field: 3-D Linked Data (graph) traversal and analysis

Field: a meta-authoring environment

Linked Data: large graphs from the web

Case Study: Field v. relFinder

Field: chart interface

Field: the Web – 3D browsing

The Field / Javascript / Web-browser bridge

Field has full support for remote, web-browser-hosted Javascript execution.

• The most immediate benefit seems to be development speed. Unlike traditional Javascript environments the Online Plugin for Field allows Javascript programs to be executed and constructed piecemeal — built up and tuned interactively — while keeping the execution of the Javascript online and in web-browsers.

• Unlike Field’s previous language and runtime bridges, in the online case multiple execution hosts may be available — each web-browser visiting a page being served by Field’s new embedded web server.

• This allows the Field to be used as a complete development environment for the authorship of online visualizations that fit directly into the Linked Open Data demo model.

• Field sources Linked Data from LOGD via Sparql, processing it, and sending it to an interactively authored Javascript visualization in Protovis.

The current version of Field running on Ubuntu 10.10

Linked Data, local navigation examples: Original and first traversals

Deeper in the graph exploring

Final traversal with relations

Step by step creation of a Field-based environment for visualizing relationships found within semantic web data. The goal is to construct a workbook that shares much of the functionality of “relFinder” — an open source, and well crafted online example of such a tool. The crux of this product is a force-directed visualization of the graph-based results of SPARQL queries against Semantic web data resources. Purpose:

• Firstly, its simply serves as a good example of many of the fundamentals of Field — drawing, interaction, “live coding”, introspection and integration with large existing libraries,

• Secondly, and more importantly, it’s an excellent example of the core argument of our research here, that by getting the fundamental abstractions right in a broad, general purpose environment like Field we can but tailor made environment like relFinder extremely rapidly.

Fielding:

Ultimately, our tutorial guides the reader through less than a page of code and would be comfortably within an afternoon workshop format. Yet this result is a tool with arguably the same or greater functionality than relFinder — which weighs in at some 15,000 lines of code after 3 years of development. The resulting implementation inside Field is radically shorter, but it’s also radically transparent — the contributions, difficulties and opportunities of the research present in the original work that we’ve duplicated here are much more visible. This tutorial suggests that our original argument that Field can offer this field an interfacing for scholarship and an “environment for constructing environments” are sound and that we can go some way to proving this by reusing existing projects inside Field.

The relFinder example shows Field implementing a common forced directed graph layout algorithm. This is an extremely commonplace layout technique, but one thing that Field’s graphics system can reach with relative ease that other runtimes essentially cannot, is GPU accelerated versions of these algorithms.

• We have written a prototype layout implementation that remains interactive at vastly higher node counts than GPU implementations.

The lack of development environments for interdisciplinary re-search conducted on large-scale datasets hampers research at every stage.

• Projects incur large startup costs as disparate infra-structure is assembled; experimentation slows when software components and environment are mismatched for specific re-search tasks; and findings are disseminated in forms that are hard to examine, learn from, and reuse.

• Behind these problems is a common cause — the lack of good tools.

• When large, heterogeneous and distributed data is added to the equation, further frustration, at the least, ensues.

• As a result using existing platforms, the programmers of 21st century interactive visualizations are reduced to working in the same fashion with the same tools as 20th century database programmers.

• Our contribution is to bring the tools of digital artists to bear on the aforementioned data analysis and visualization challenges.

• Here we report on the current state of progress in adapting Field for large-scale, web-based scientific data analysis and visualization with an emphasis on Linked Open Data and especially the current data hosted by RPI.