Delta.js - A Diff and Patch Engine for DOM Trees

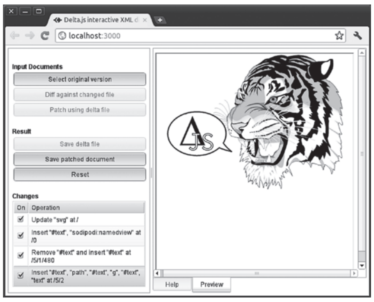
Web and Business Applications / Advisor: Prof. Dr. Olivier Biberstein
Expert: Peter Matti

Delta.js is a JavaScript framework capable of detecting changes between two versions of a structured document. Furthermore it provides methods to apply those changes back to the original document or a slightly modified version thereof. Our implementation runs in any modern web browser as well as on JavaScript based server software like node.js and on the command line. HTML and any XML based document formats are supported, including SVG.

Application Independent Version Control

When different authors are working on the same document, the ability to track and manage changes is essential. Existing solutions often are tied to the editor application, i.e. the word processor or spreadsheet software. As a consequence the involved editors are forced to use the same applications. This approach works fine for small teams collaborating in a managed office environment. But such a restriction might result in major problems when a workflow requires integration of third party systems.

In the software industry, distributed teams rely heavily on generic version control systems, which are capable of tracking and merging changes to source code. Those systems work independently of the tools and editors used by the programmers, as long as the source code is kept in plain text files.



On	Operation
☑	Update "svg" at /
☑	Insert "#text", "sodipodi:namedview" at /0
☑	Remove "#text" and insert "#text" at /5/1/480
☑	Insert "#text", "path", "#text", "g", "#text", "text" at /5/2

Realtime preview when applying a set of changes to an SVG image.

Because more and more vendors are adopting XML as the basis of their own file formats, a more generic approach to change management of structured documents independent of the editor application becomes realistic. The algorithms implemented in Delta.js may form the basis for a change management system for structured documents.

Diff and Patch for Structured Documents

A number of algorithms suitable for detecting changes in structured documents have been studied for this thesis. Based on our research, we›ve chosen XCC (XML Change Control) recently released by Sebastian Rönnau and Uwe M. Berghoff as the starting point for our own implementation. The main assumptions of the chosen approach are:

- Document order is important.
- Interesting content is contained within leaf-nodes.

While we maintained the basic concepts of the XCC-diff algorithm, our version departs significantly from the original implementation in some parts. As a result the runtime of our implementation is easier to predict in some cases.

XCC also proposes a file format suitable for storing a set of changes into a file, commonly referred to as a patch. Along with the changed content, a context fingerprint representing the neighborhood of where the change took place in the original document is recorded in the patch. When a patch is merged back into the original document or a modified version thereof, the patching algorithm is capable of verifying and adjusting the location of a change by comparing nearby content with the context fingerprint.

In order to demonstrate the capabilities of the Delta.js framework we provide a set of browser based JavaScript applications. When supplied with two versions of an XML file, our application computes the changes between them. Then the original source code is displayed and changed parts are highlighted. For certain file formats like HTML and SVG a life preview is available where the impact of individual changes can be monitored in realtime.

Delta.js is publicly available from http://github.com/znerol/node-delta

Lorenz Schori