MICHIGAN STATE
UNIVERSITY

Project Plan

Optimizing Firefox Localization

The Capstone Experience

Team Mozilla

Avery Berninger
Brain Chen
Chris Frey
Ian Kirkpatrick
Nicholas Cowles
Yuan Cheng

Department of Computer Science and Engineering
Michigan State University

Spring 2019

From Students…
…to Professionals

Functional Specifications

- Competition among browsers
- Continued migration to Fluent
  - better user experience
- Better developer experience
  - tests mitigate errors from repetitive tasks
- Gauging performance gains

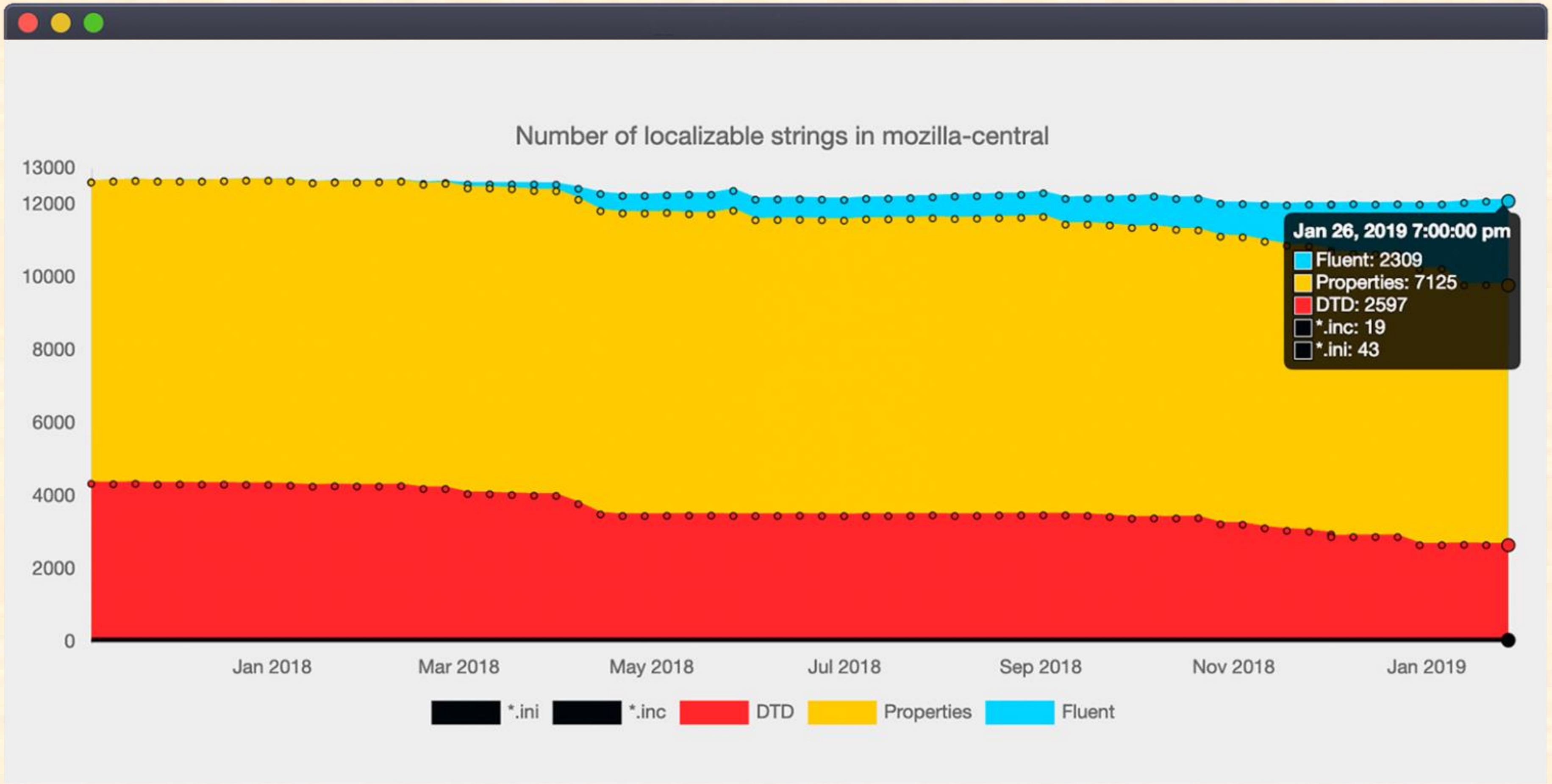
Design Specifications

- New translation paradigm
  - asymmetrical translations

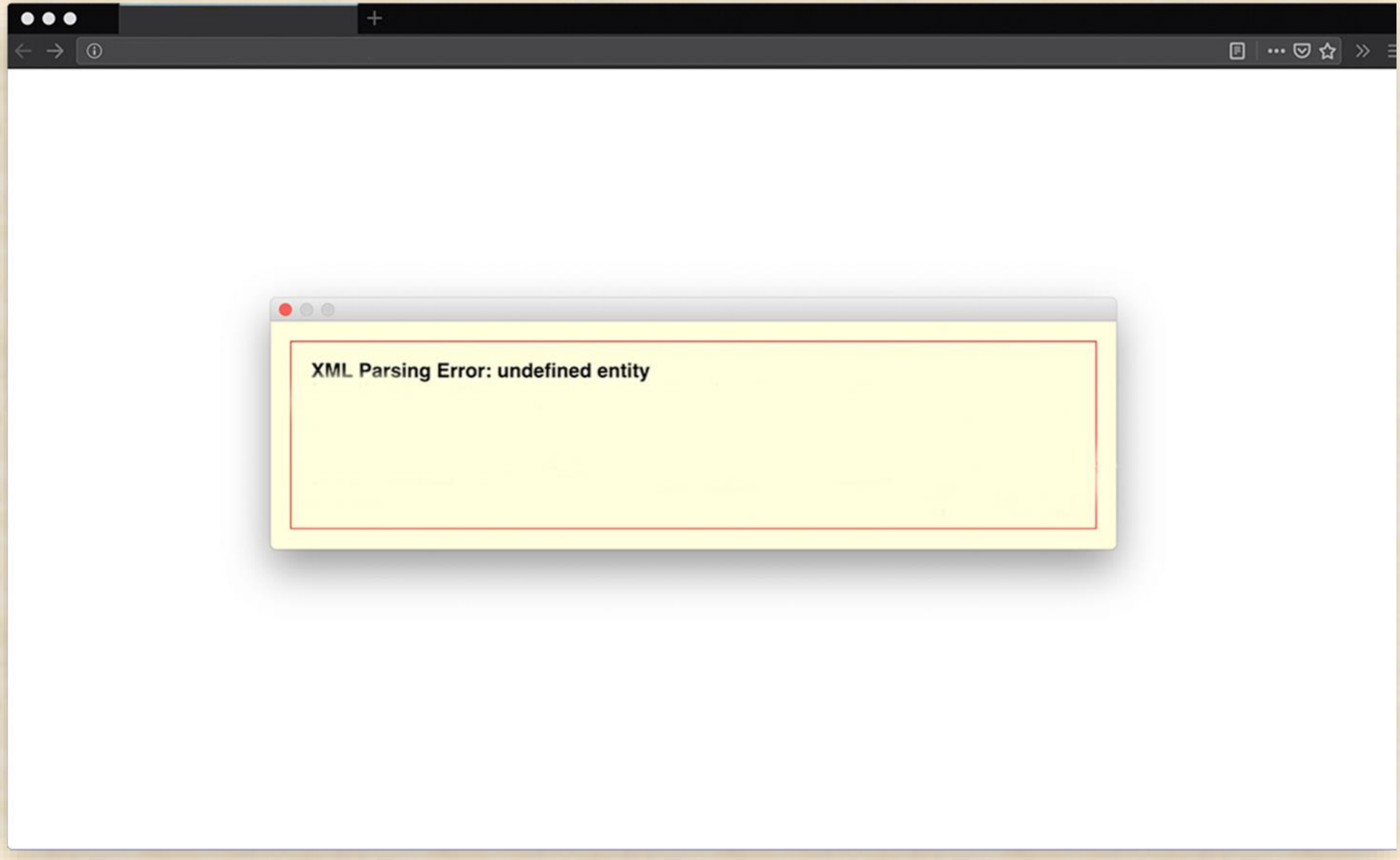
- Faster interface
  - faster loading with WebAssembly

- Extended tooling around Fluent

Screen Mockup: Fluent Migrations

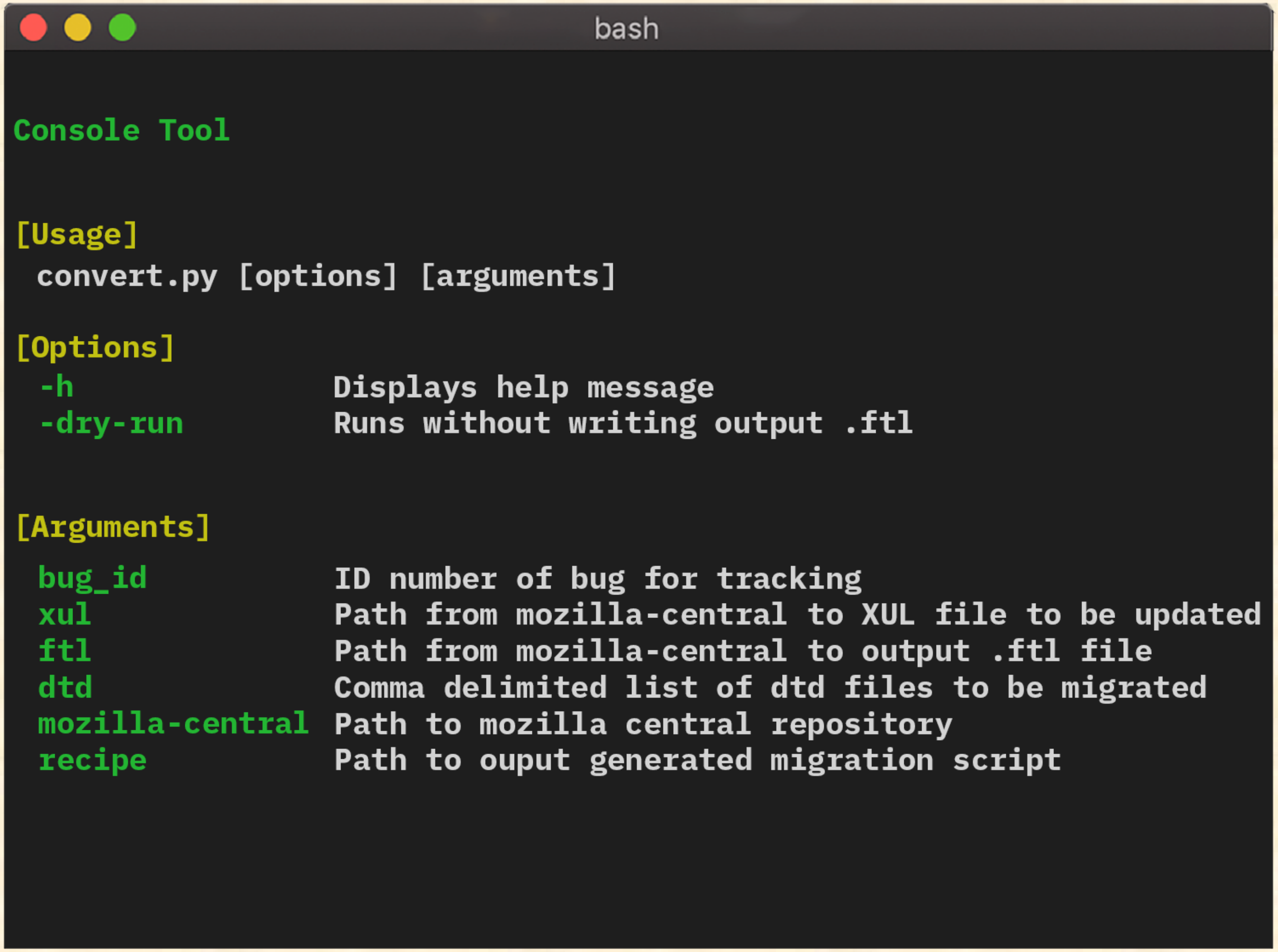


Screen Mockup: Yellow Screen of Death



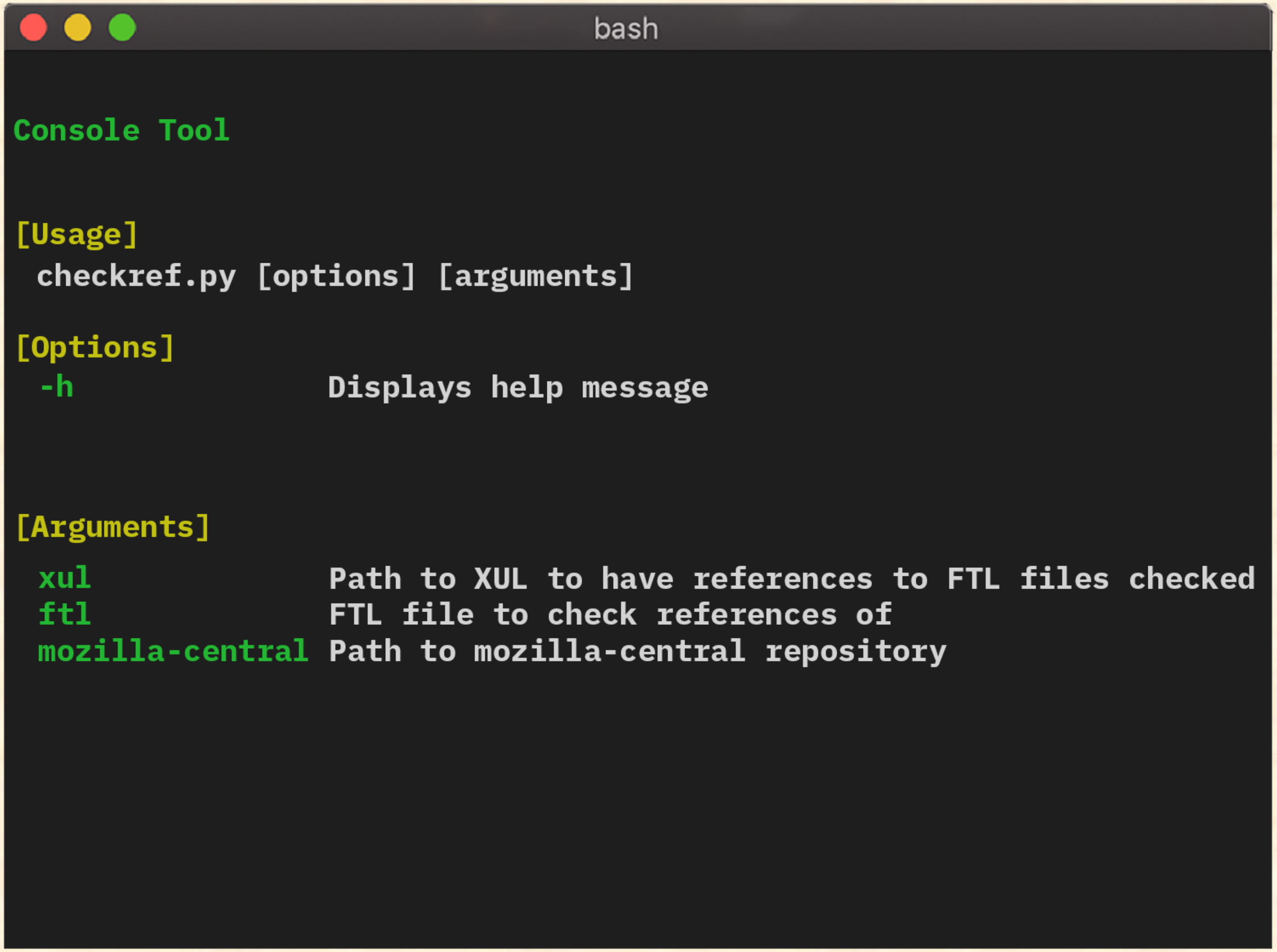
XML Parsing Error: undefined entity

Screen Mockup: Console Tool - Conversion



```
Console Tool


[Usage]
 convert.py [options] [arguments]

[Options]
 -h               Displays help message
 -dry-run         Runs without writing output .ftl


[Arguments]

 bug_id           ID number of bug for tracking
 xul              Path from mozilla-central to XUL file to be updated
 ftl              Path from mozilla-central to output .ftl file
 dtd              Comma delimited list of dtd files to be migrated
 mozilla-central  Path to mozilla central repository
 recipe           Path to ouput generated migration script
```

Screen Mockup: Console Tool - Reference Checking



```
Console Tool

[Usage]
 checkref.py [options] [arguments]

[Options]
 -h              Displays help message



[Arguments]
 xul             Path to XUL to have references to FTL files checked
 ftl             FTL file to check references of
 mozilla-central Path to mozilla-central repository
```

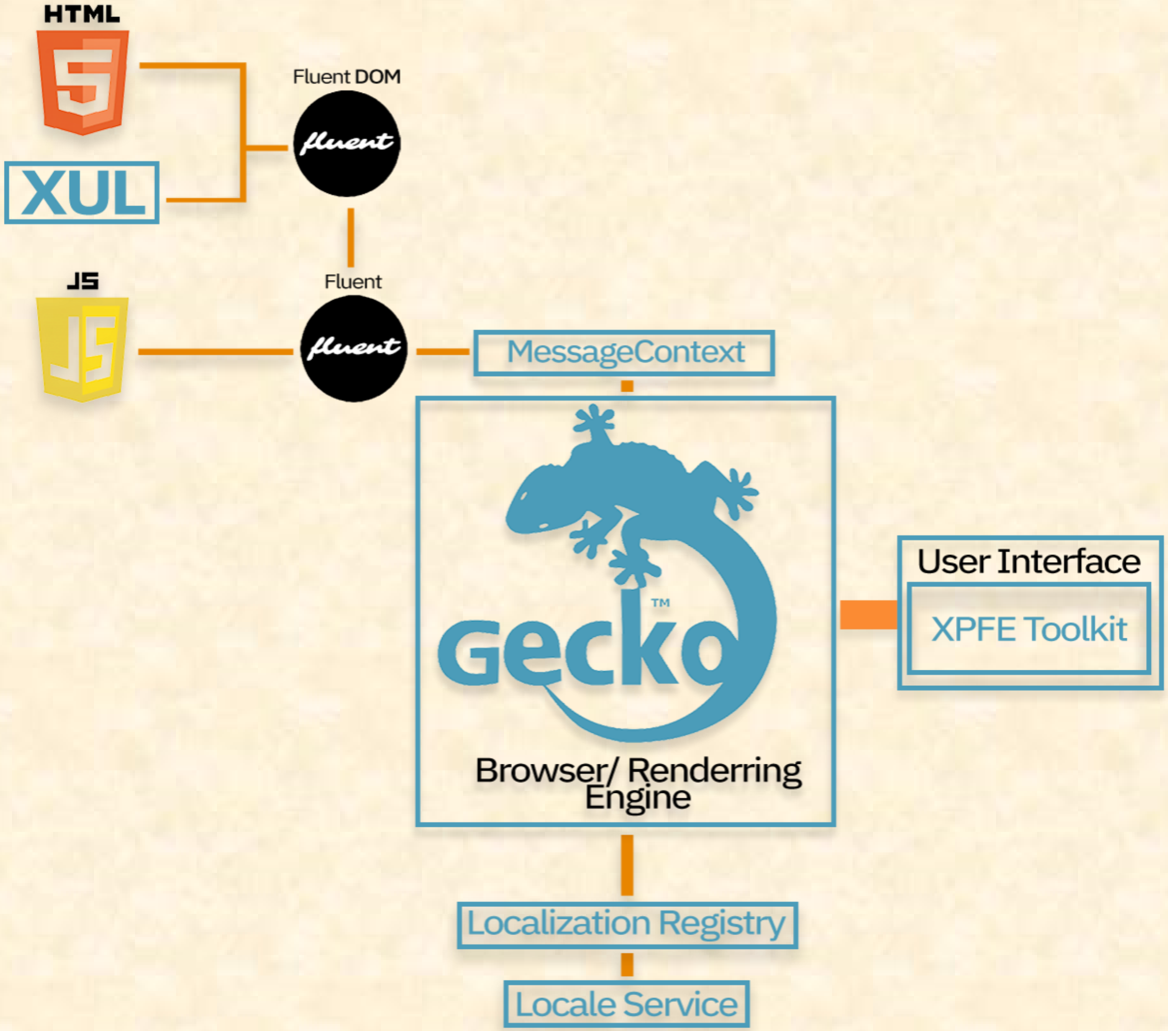
Technical Specifications

- Software Technologies
  - Python
    - Used to develop migration scripts for translating localization files from .dtd to .ftl (Fluent) format.
    - Converts variables and associated translations from .dtd into a new .ftl file.
    - When parsing invalid or incomplete .dtd files from our localizers, Firefox crashes with the “Yellow Screen of Death”. Fluent will help fix parser-based browser crashes.
  - JavaScript
    - Currently used as a part of a Fluent-based text-parser for web pages.
    - When attempting document localization, this parser searches through a document and localizes all specially marked elements.
    - Currently implements a Promise system for asynchronous translation.
  - Rust
    - Improvement upon currently implemented JavaScript-based text parser.
    - Will be transpiled into JavaScript using WebAssembly.
    - Development of Rust-based text parser in progress. Will theoretically speed up text parsing.
  - XHTML, XML & XUL
    - Mark-up and binding languages used to define the browser and user interface.
    - Uses localization information from .dtd and Fluent files to properly localize the UI.
  - Fluent
    - Tool used to improve software localization.
    - Provides an effective method for mapping complex language translations.
    - Employs modularity, simplicity, and composability for tandem use with other technologies.

Technical Specifications

- Testing Technologies
  - Mochitest Framework
    - Automated testing framework build on top of MochiKit JavaScript Libraries.
    - Used to locally create and run browser tests by simulating user activity at high speeds.
    - Displays test success or failure through the use of JavaScript function calls.
  - ESLint Engine
    - Uses a set of configuration files loaded within the Firefox codebase so JavaScript can be analyzed for errors.
    - Comprised of a set of rules which analyzes code for correctness and style consistency.
  - Try Server
    - Used to remotely test a patch before checking it into the core Firefox repository.
    - Tests patch to determine correctness and identify bugs.
    - Can view a list of issues with patch using treeherder.mozilla.org.
  - Python UI Test
    - Similar to Mochitests, yet designed to be more powerful.
    - Can restart the browser within the test.

System Architecture



HTML

XUL

Fluent DOM

JS

Fluent

MessageContext

Gecko

Browser/ Renderring Engine

User Interface

XPFE Toolkit

Localization Registry

Locale Service

System Components

- Hardware Platforms
  - Microsoft Windows
  - MacOs
  - Linux
  - Android/IOS
- Software Platforms / Technologies
  - MSYS based terminal for building Firefox on Windows
  - Mercurial VCS
  - Bugzilla ticket tracker
  - Phabricator for code review
  - Mozilla Try server and Mochitests
  - Visual Studio Code, vim, emacs, notepad++ text editors
  - Searchfox

Risks

- Large Codebase
  - An enormous code base of more than 35 million lines of code written in many programing languages
  - use searchfox.org, communicate with Firefox developers
- Asynchronization
  - The previous localization system had always translated strings synchronously
  - Searchfox and the tests run through the testing framework
- Tooling
  - Current tooling around the task of migrating the past localization system to Fluent is sparse and much has to be done by hand
  - Talking with developers involved in migrating Firefox to using Fluent and seeing what their greatest needs are
- Benchmarking Rust based Fluent Parser
  - Test the viability of a new parser written in Rust and transpiled into WebAssembly
  - Sophisticated telemetry tools

Questions?

? ? ? ?

? ?

? ? ?