Loom: One Web Development Tool for Everyone

Kyle Cutler (kjc9861@rit.edu)

Advised by Dr. James Heliotis

Background

Existing web development tools either provide easy-to-use interfaces intended for novice users, sacrificing expressiveness, or advanced interfaces intended for developers who already know web technologies, but inaccessible by ordinary users.

Loom aims to bridge the gap between these categories, giving novice users the full power of web technologies such as HTML and CSS without the need to write code, while also giving developers a powerful tool for rapid prototyping.

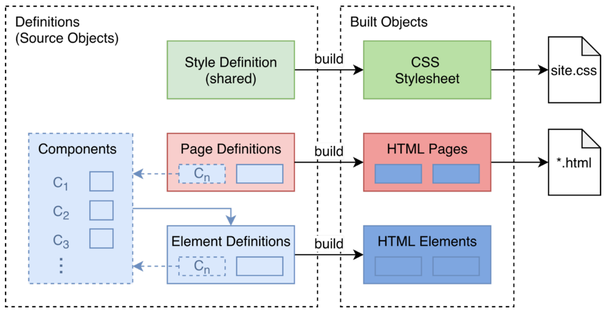
In addition, Loom aims to impart the basics of these technologies to users as they work with the tool, teaching valuable concepts that can be carried over to other web development tools.

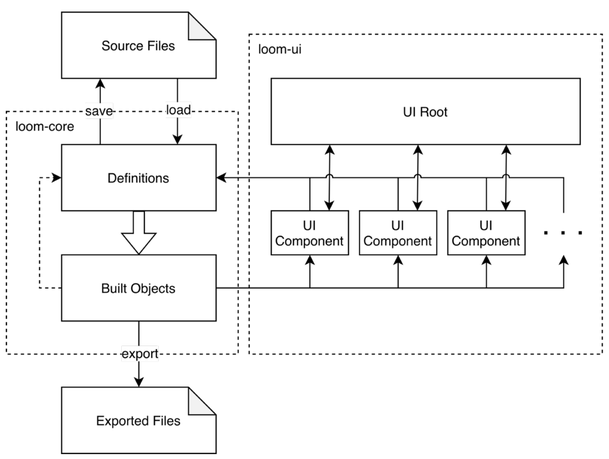
Solution Design

A Project in Loom can consist of several pages and one stylesheet which contains styling information for the entire project.

Working directly with web technologies can be frustrating, even for developers. Loom provides a handful of simple, but useful, abstractions over these technologies to make working with them easier.

Components are an example abstraction defined by Loom. Components enable reuse of HTML elements across several pages, such as a site header and footer. This is not possible in standard HTML.





To support these abstractions, Loom differentiates between Definitions and Built Objects. Definitions are the data which users directly modify. Built Objects result from building Definitions, and correspond directly to standard HTML and CSS.

The build process to transform Definitions into Built Objects is "reactive". That is, if any value within a Definition changes, any associated Built Objects will update appropriately as well, and also emit events to trigger additional actions such as UI updates. This enables immediate feedback resulting from user actions.

Loom makes heavy use of event-driven programming. A collection of event-driven data structures created specifically for Loom enables a high level of code reuse.

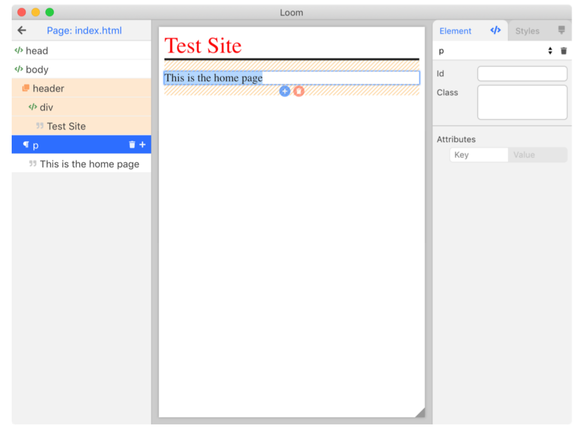
Interface Design

The UI of Loom is divided into three main sections: the Navigator, the WYSIWYG Editor, and the Properties Editor.

The **Navigator** (left) allows the user to select the page or component to edit, and then a data item within.

The **WYSIWYG Editor** (center) shows the user a live preview of the page and selected data, and allows them to edit the page contents via direct text entry, keyboard shortcuts, and floating controls.

The **Properties Editor** (right) gives the user control over properties of the selected data such as HTML attributes and styles.



Results

To evaluate Loom, we evaluate how well it meets the needs of target users: Novices, Developers, and Designers.

Novices	Developers	Designers
✓ Create / Delete Pages	✓ Set Custom HTML Attributes	✓ Create Content Quickly
✓ Create / Delete Elements	✓ View Element Properties	✓ Live View of Output
✓ Associate Visuals with HTML	✓ View Element Styles	○ Layout Content Quickly
✓ Save Project as Static Files	○ Set Custom CSS Properties	○ Modify Font Styles
○ Bold / Italic / Underline Text	○ Create Custom CSS Rules	✗ Copy and Tweak Pages
✗ Create Images / Links	✗ Create and Edit Scripts	✗ Standard Color Palette

There is clearly room for improvement even among this small subset of use cases. Nonetheless, this demonstrates the potential for Loom to accommodate a wide variety of users with different needs.

Future Work

UI Improvements

- Drag-and-drop to create and move content
- Additional controls within WYSIWYG Editor
- Improved interface for editing style properties
- Additional context clues, help text
- More…

Additional Features

- Parameterized components
- Support for media files, scripts
- Import existing websites
- Post-processing of output content
- More…