Web Extensible Display Manager 2*

R. Slominski, T. Larrieu
Jefferson Lab, Newport News, VA 23606, USA

Introduction

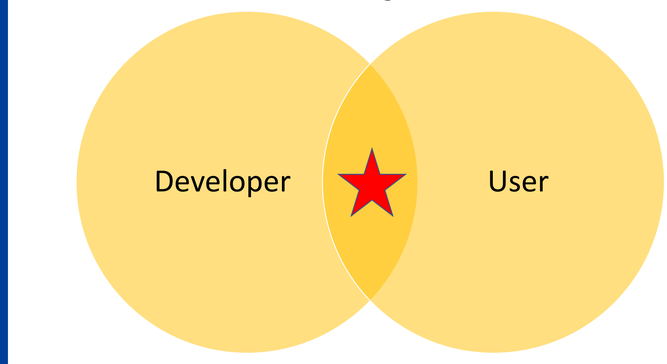
The Web Extensible Display Manager (WEDM) was first deployed at Jefferson Lab (JLab) in 2016 with the goal of rendering Extensible Display Manager (EDM) control screens on the web for the benefit of accessibility, and with version 2 our aim is to provide a more general purpose display toolkit by freeing ourselves from the constraints of the EDM dependency.

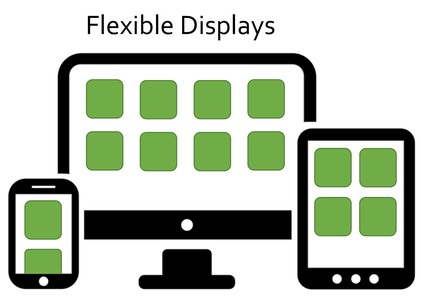
Version 2.0

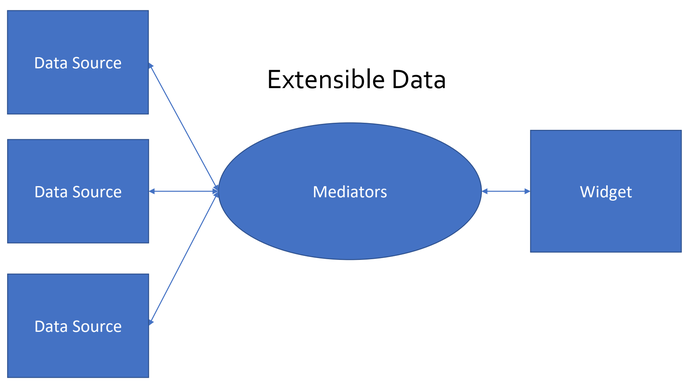
Version 1.0

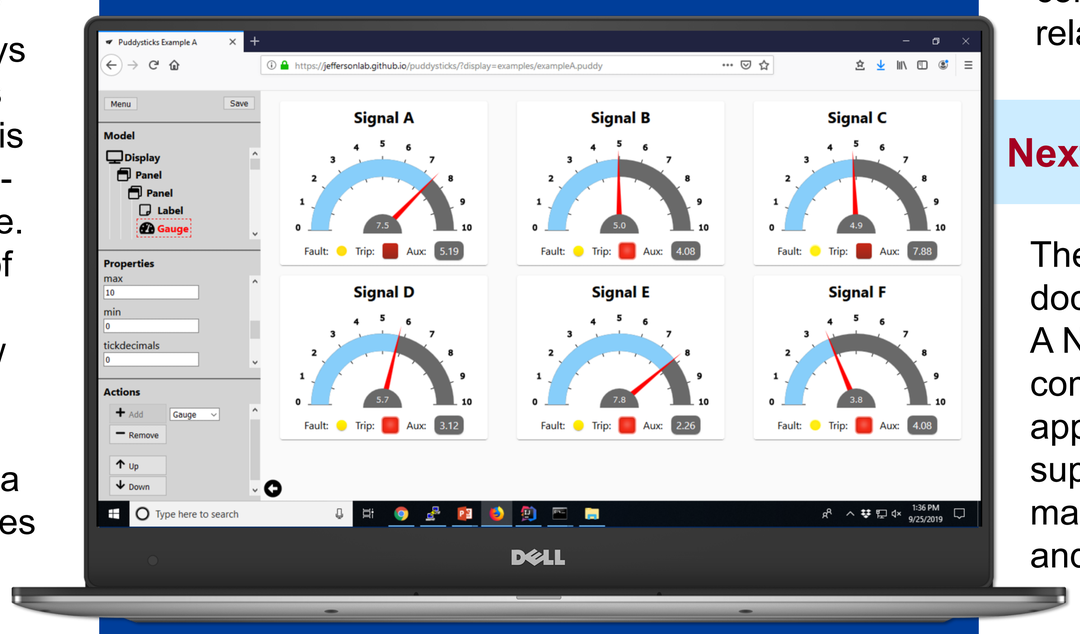
Background

There has been an explosion of interest in control system displays on the web in the last few years. It has increasingly become a user expectation for displays to be available on a variety of devices including smart phones, and the web is a well-established standardized cross-platform way to deliver this experience. Control systems have a long history of display managers: Graphical User Interface (GUI) builder tools that allow non-programmers to create control screens without writing any code. We report our status as we work towards a web based display manager that serves all devices, whether they be in the control room or in your pocket.









Towards a Better Display Manager

Web technologies provide an established and powerful application environment, yet unique challenges arise. On the web an overwhelming number of frameworks exist to attempt to ease application development effort, sometimes at end-user expense. In order for displays to render properly on screens of various sizes they must be intentionally designed for such use. Creating reusable widgets is facilitated by frameworks, but also requires careful code design to avoid tight coupling of data.

Puddysticks

SVELTE

The second version of WEDM is named Puddysticks. This prototype software provides a web based display builder tool, and an initial basic set of widgets including a Panel, Label, Indicator, and Gauge. This proof-of-concept creates and consumes display files in JavaScript Object Notation (JSON) format, and is built on Svelte, a reactive compile time component framework. Integration with EPICS is provided out-of-the box, and extension to other data sources is encouraged.

Solution

Compiled Reactive Framework

Both developer convenience and fast user experience can be achieved by compiling developer friendly reactive code into concise imperative quick to download and execute browser code.

Responsive Web Design

A single display can be reused on devices of varying screen size if designed to be responsive. Users can focus on grouping and ordering widgets and let powerful browser layout engines do the rest via Cascading Style Sheet (CSS) rules such as grid and flexbox.

Pluggable Data

Widgets are decoupled from their data source using the Mediator design pattern to encourage integration with different control systems such as EPICS or even relational databases or web services.

Next Steps

The source code, demo, and documentation is available on GitHub. A NodeJS build script is provided for compiling and packaging the application. Puddysticks is a useful supplement now, but before replacing a mature display manager authentication and input widgets must be added.

Download at:
https://github.com/JeffersonLab/puddysticks

Conclusion

Puddysticks demonstrates a promising web based display manager that makes remote monitoring easy, now without an EDM dependency. We used a compiled reactive framework to balance user and developer ergonomics, responsive web design to allow a single display to be reused on multiple devices, and pluggable data sources to enable extensibility to control systems and web services. The software is a useful supplement now, but authentication and input widgets are needed before it can replace a production display manager.

*Authored by Jefferson Science Associates, LLC under U.S. DOE Contract № DE-AC05-06OR23177