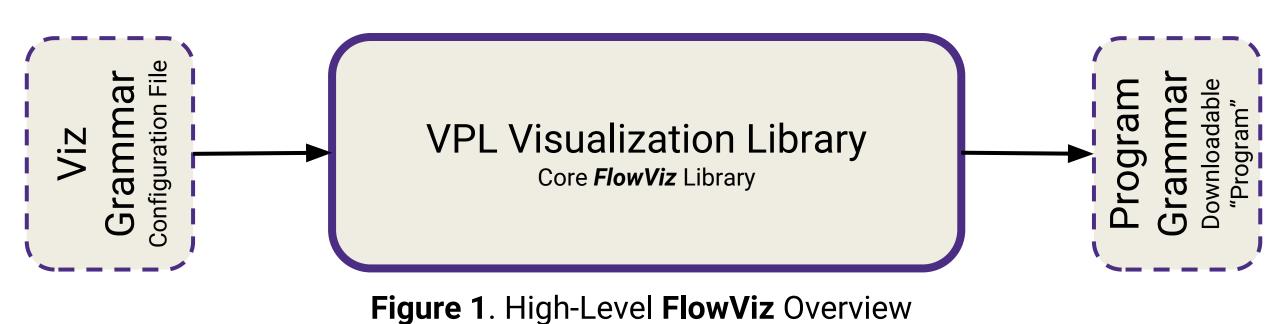


# **FlowViz**: a Visualization Toolkit to Support Visual Programming Language Development

Alexander Fiannaca and Sonya Alexandrova

## Introduction

Many end user programming systems consist of visual editors which allow users to create data flow graphs representing the logical flow of the program structure. Unfortunately, there are no unified tools for creating new visual programming languages (VPL) and their IDE's. Therefore, we introduce *FlowViz*: an interactive visualization toolkit that facilitates the creation of new VPLs.



# **Related Work**

Existing Tools: only allow creation of programs in a specific VPL



Figure 2. Examples Popular Existing Visual Programming Languages

<u>Viz Libraries</u>: used to draw graphs, but tedious when creating VPL's as this approach requires writing lots of boilerplate code for each VPL.



Figure 3. Visualization Libraries That Could Be Used to Implement a VPL

# **Library Design & Implementation**

FlowViz was implemented in Node.Js via Browserify and makes heavy use of D3.js, Snap.svg.js, and a series of core Node modules.

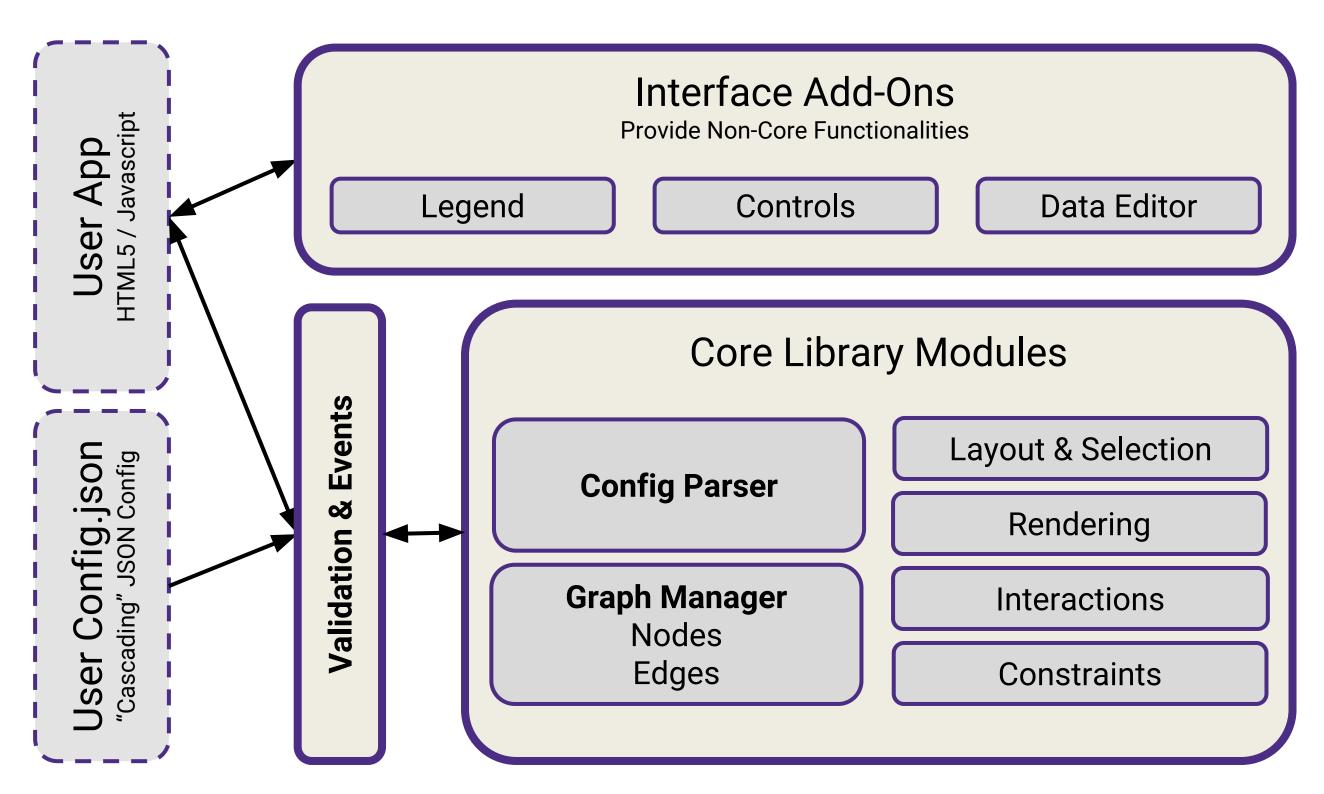
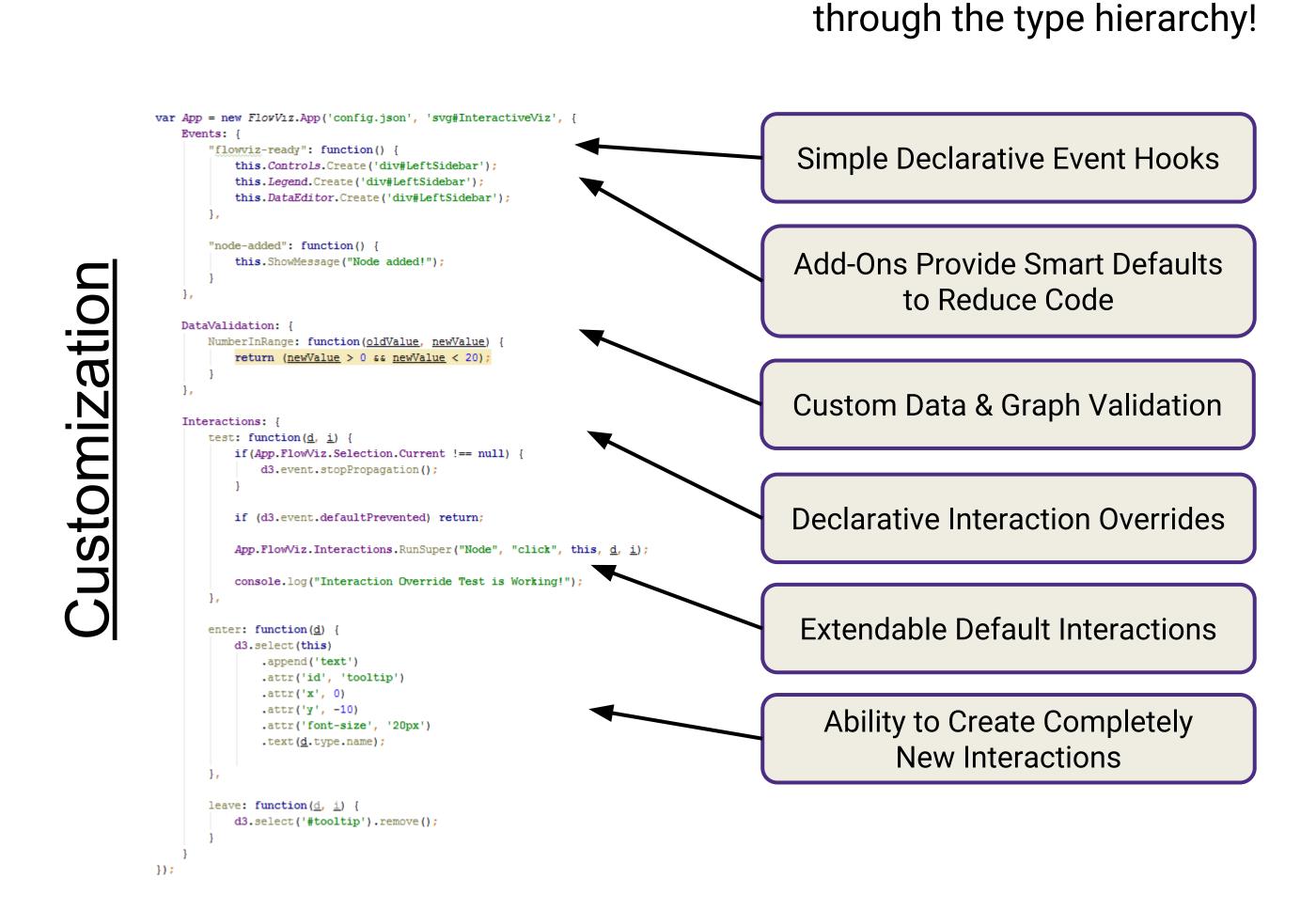


Figure 4. FlowViz Library Design

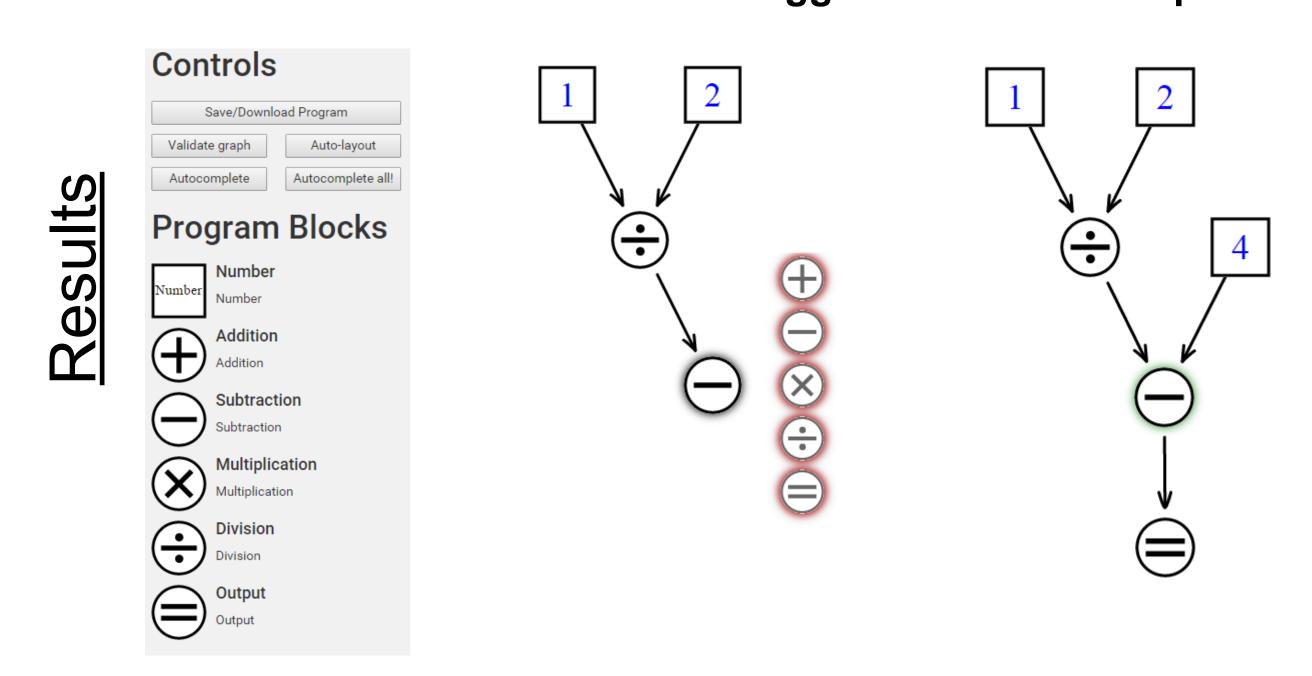
#### **Details of System Use** Set Display Parameters 'display": {"scale": 0.5...}, 'interactions": "Node": [{"event": "click"...}, {"event": "mouseenter"...}, Override Default Interactions & "types": [{ "type": "InputDevices", **Specify New Interactions** "desc": "This is the class of all input devices", Create a Node Type Hierarchy "subtypes": [{ "view": "<?xml version=\"1.0\" encoding=\"utf-8\"?> <!--"desc": "This is a QWERTY keyboard", Attach Data to Nodes & Edges "constraints": {...} "view": "<?xml version=\"1.0\" encoding=\"utf-8\"?><!-- ( Specify Node View via SVG "incoming": {...} "outgoing": {...}

Config values cascade



Use Constraints to Define "Syntax"

### Interactive VPL IDE with Auto-Suggest and Auto-Complete!



## **Future Work**

- Edge types and custom connections based on node types
- More intelligent path planning (A\*?)
- Better set of automatic layout styles (no more Sugiyama!)
- Add unit tests