What do Researchers Need when Implementing Novel Interaction Techniques?

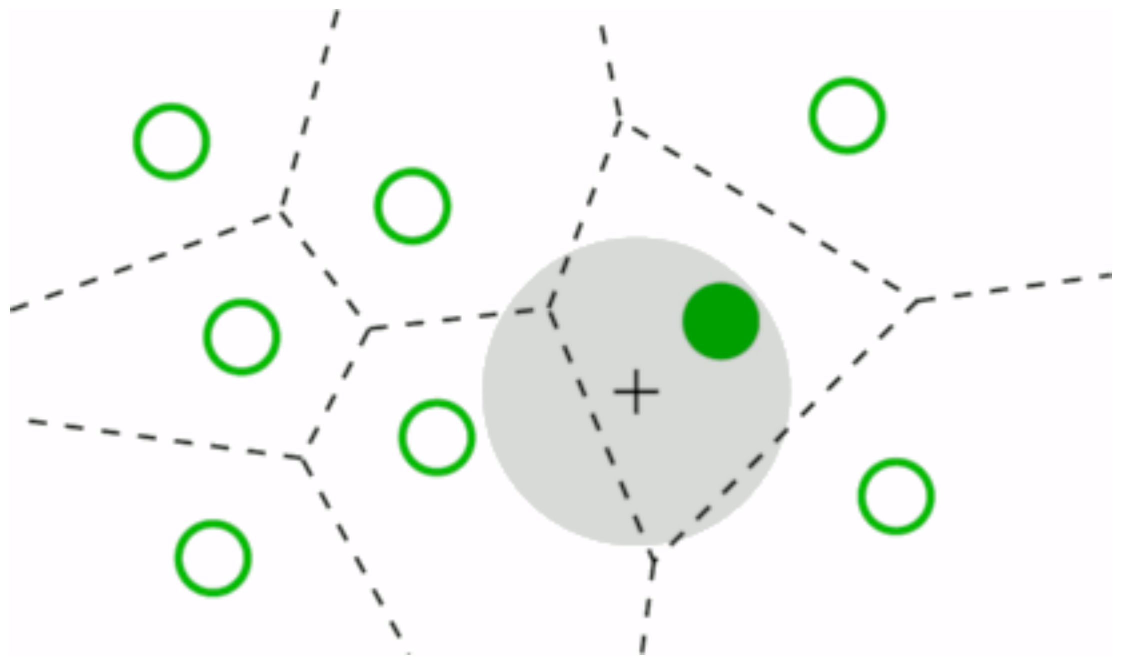
Thibault Raffaillac
École Centrale de Lyon

Stéphane Huot
Inria Lille – Nord Europe

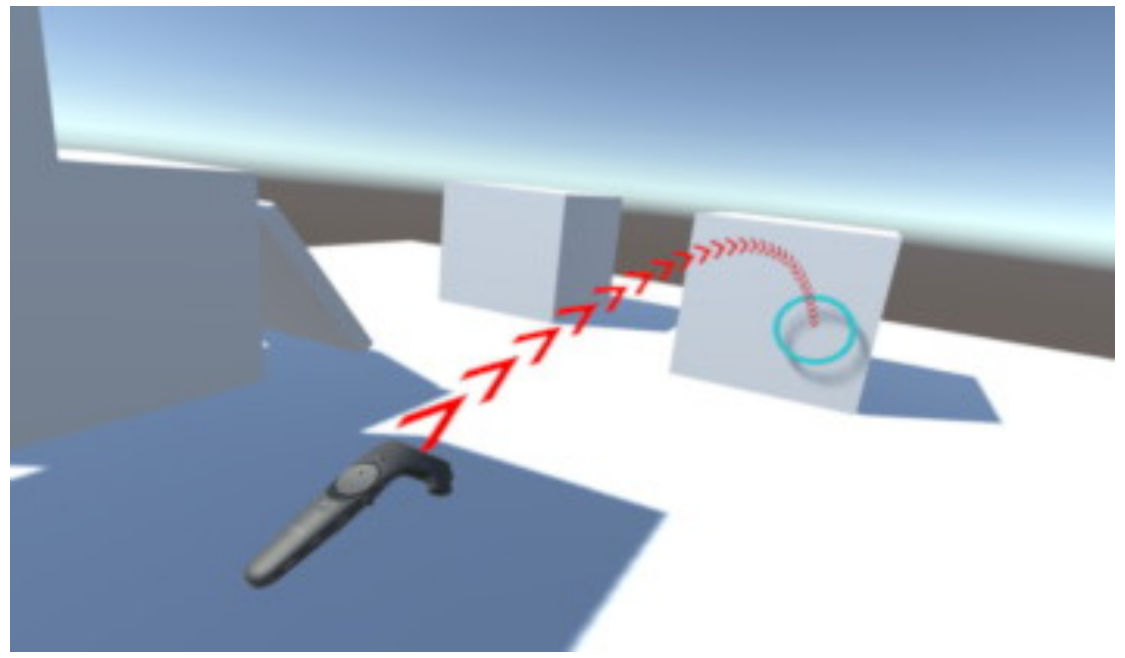
Introduction

Motivation

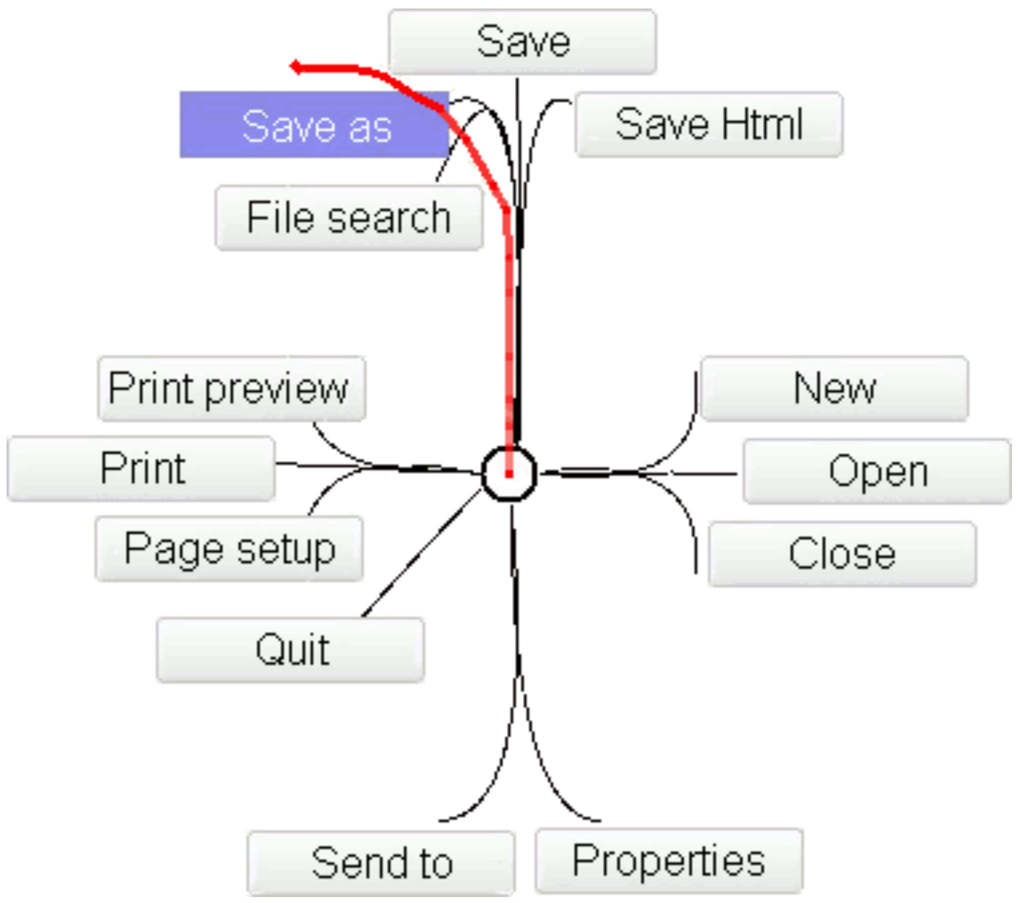
Frustration of colleagues when programming novel interaction techniques for research



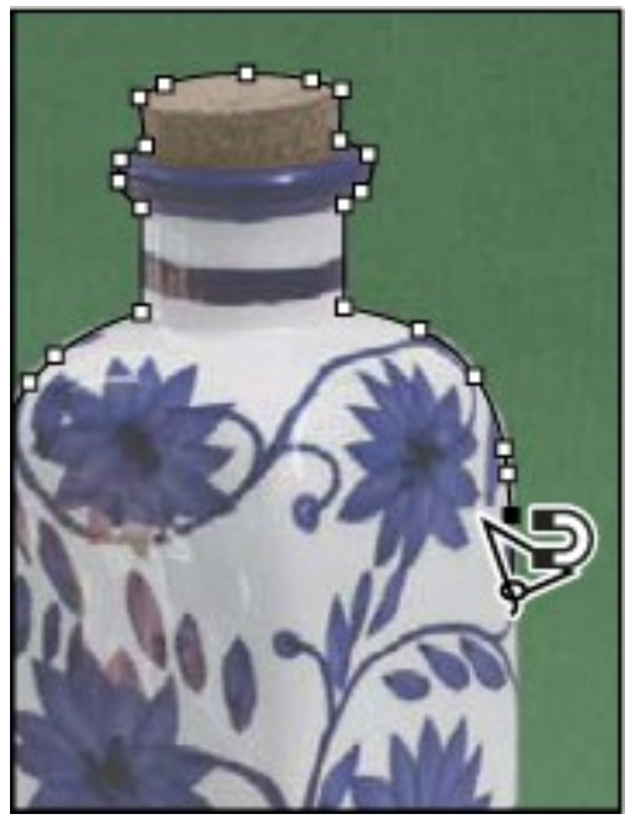
Bubble Cursor (Grossman & Balakrishnan)



Unity VR Arc Teleporter



Flower menu (Bailly et al.)



Photoshop Lasso selection

ExposeHK (Malacria et al.)

Introduction

Problem

They may use:

- an interaction framework (Qt, HTML/JS, Swing)
- a research toolkit (D3, Amulet)

Frameworks are popular but:

- input data is hard to obtain
- insufficient granularity of reuse
- unchangeable behaviors
- lagging support for new devices

Consequences:

- limited adoption of innovative interactions (trackpad, gestures, eye tracking)
- recurrent publications of tricks to circumvent limitations (Prefab, Scotty)
- active research on toolkits/architectures as alternatives to frameworks

Introduction

Plan & Research questions

- Interviews & Survey

  What do researchers do when prototyping new interaction techniques?

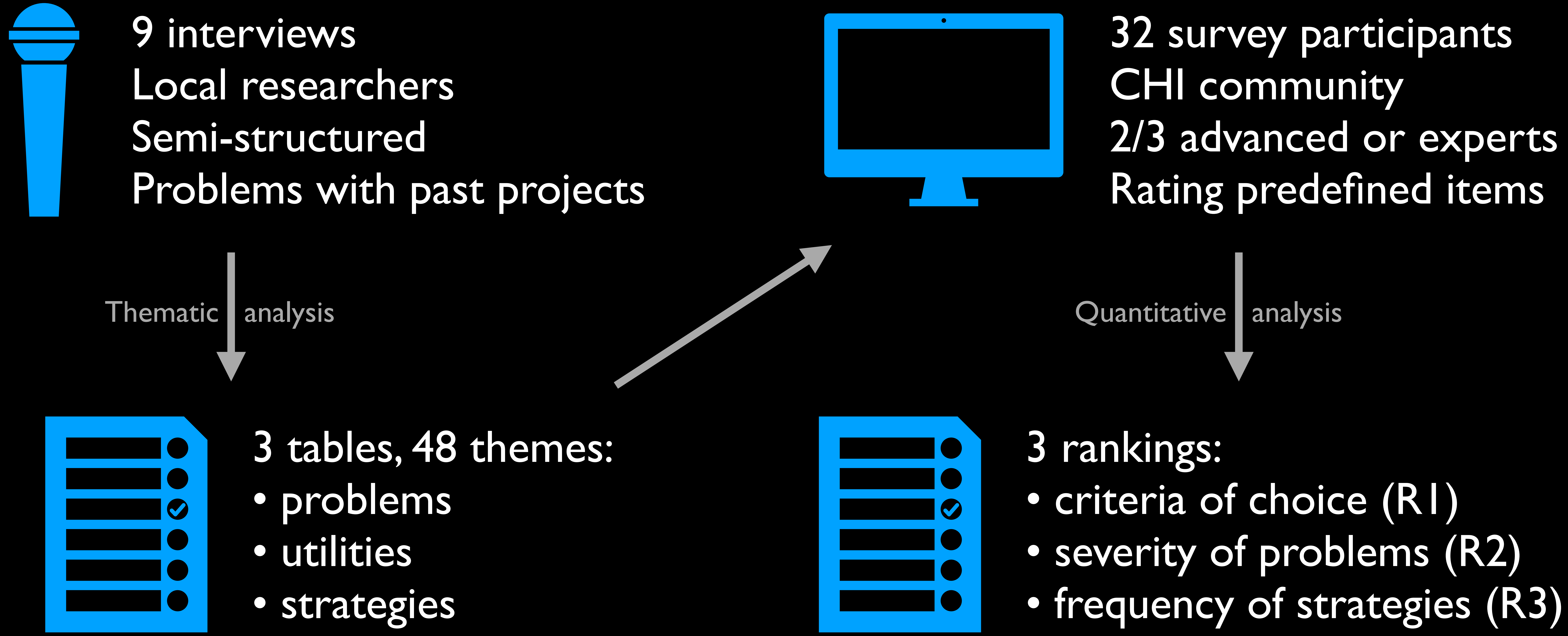
- Design recommendations

  How can we design or adapt existing frameworks and toolkits to support them?

What do researchers do when prototyping new interaction techniques?

Interviews & Survey

Methods & Analyses



9 interviews
Local researchers
Semi-structured
Problems with past projects

Thematic analysis

3 tables, 48 themes:

- problems
- utilities
- strategies

32 survey participants
CHI community
2/3 advanced or experts
Rating predefined items

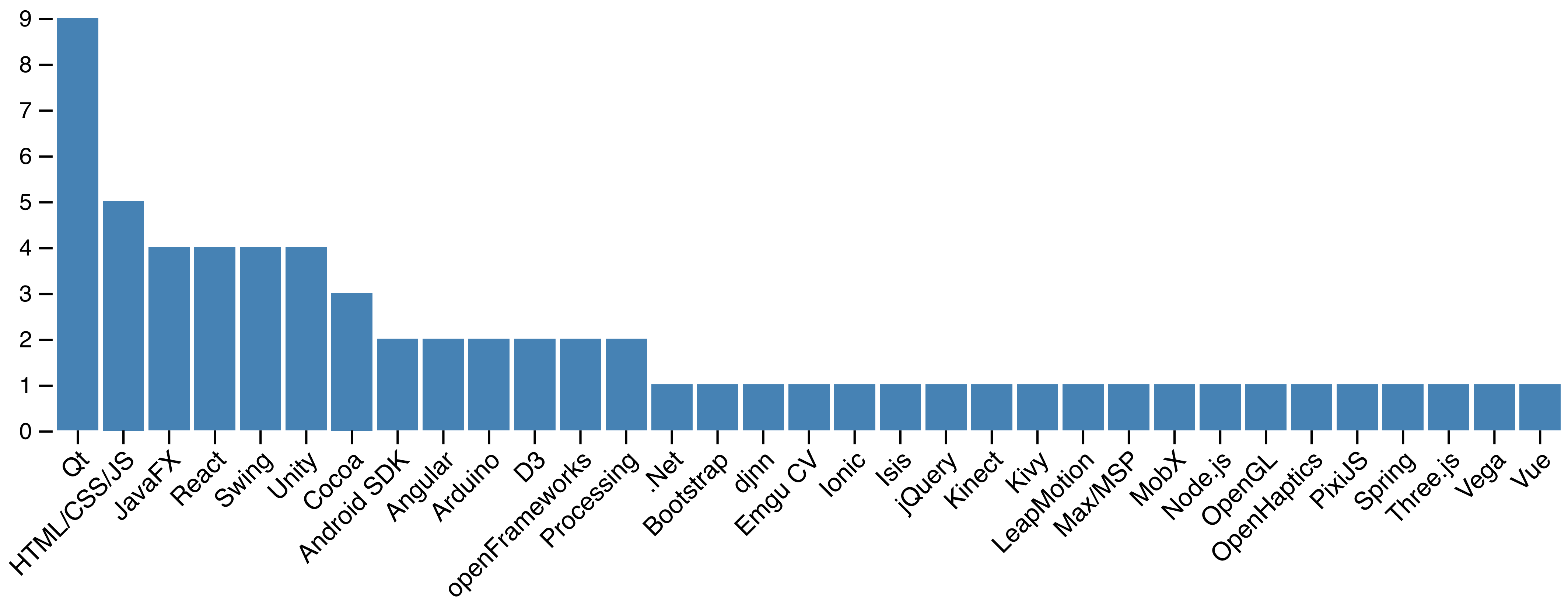
Quantitative analysis

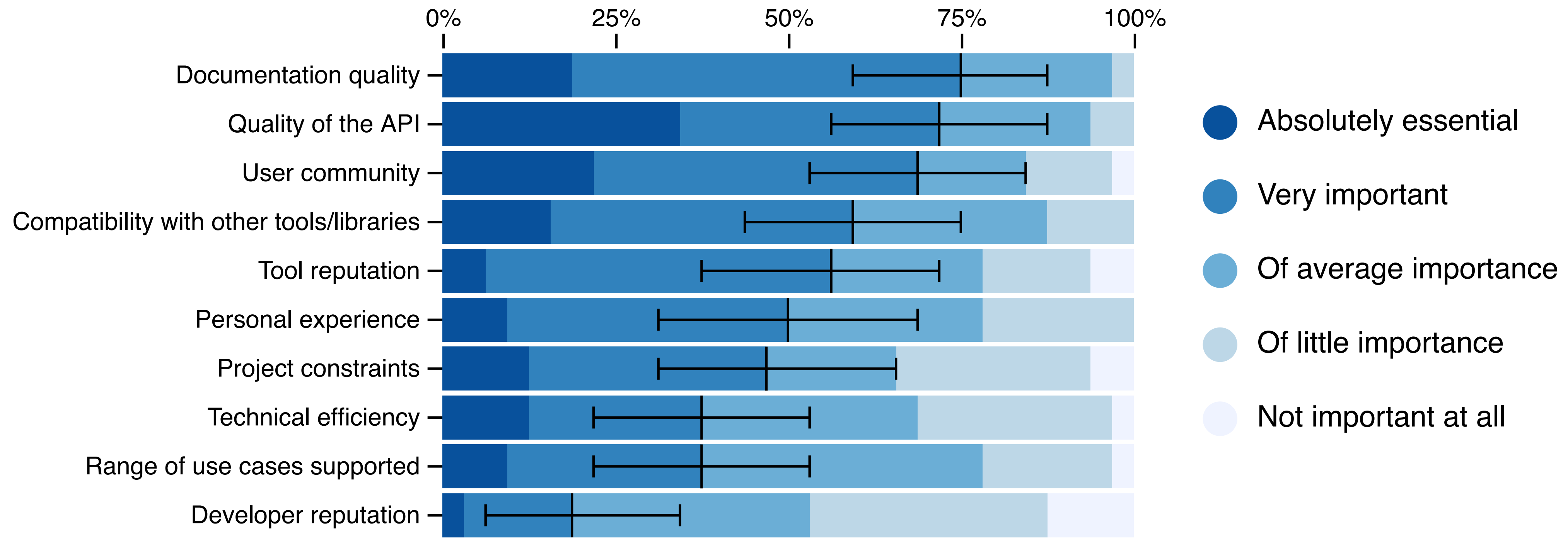
3 rankings:

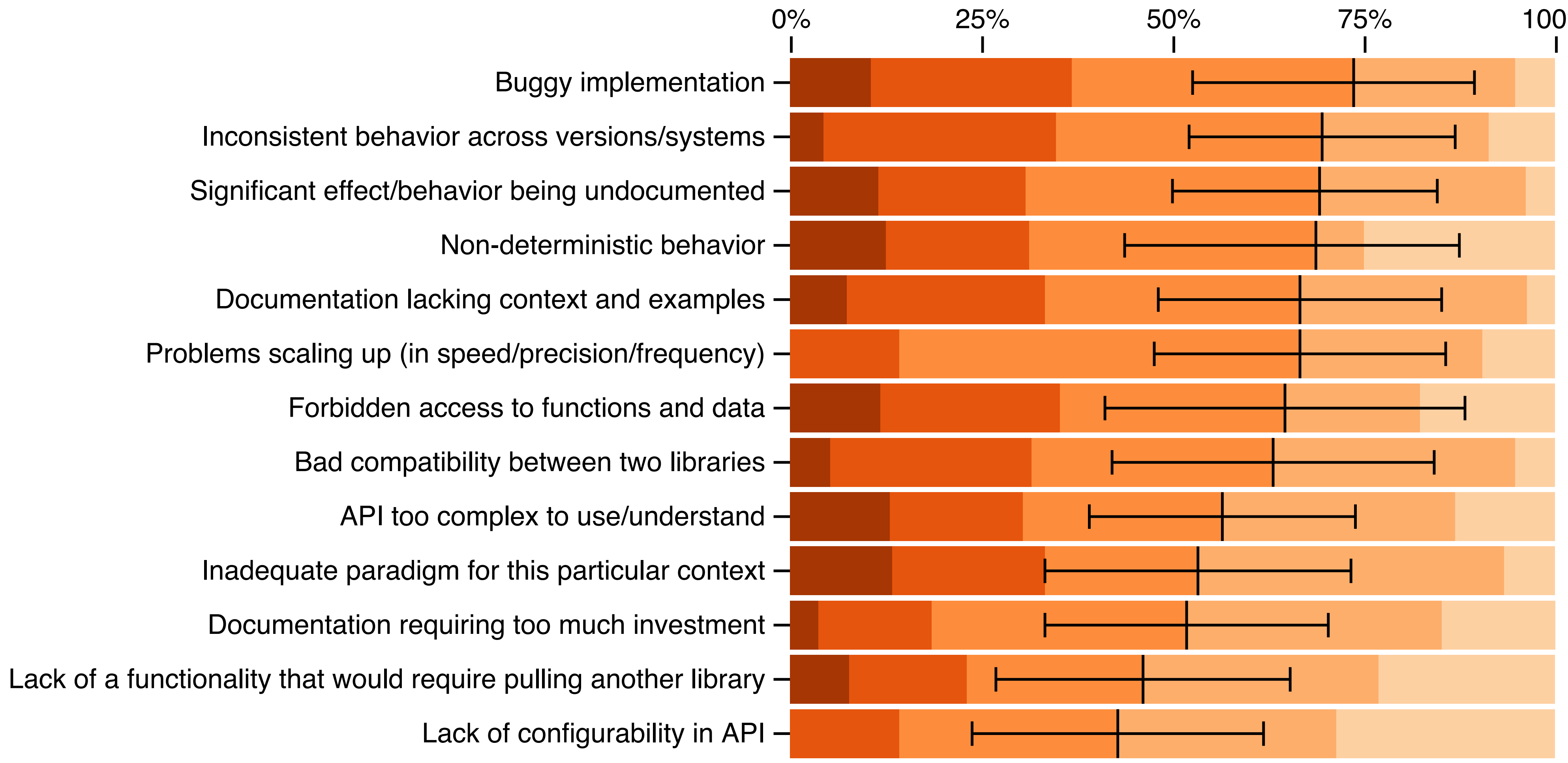
- criteria of choice (R1)
- severity of problems (R2)
- frequency of strategies (R3)

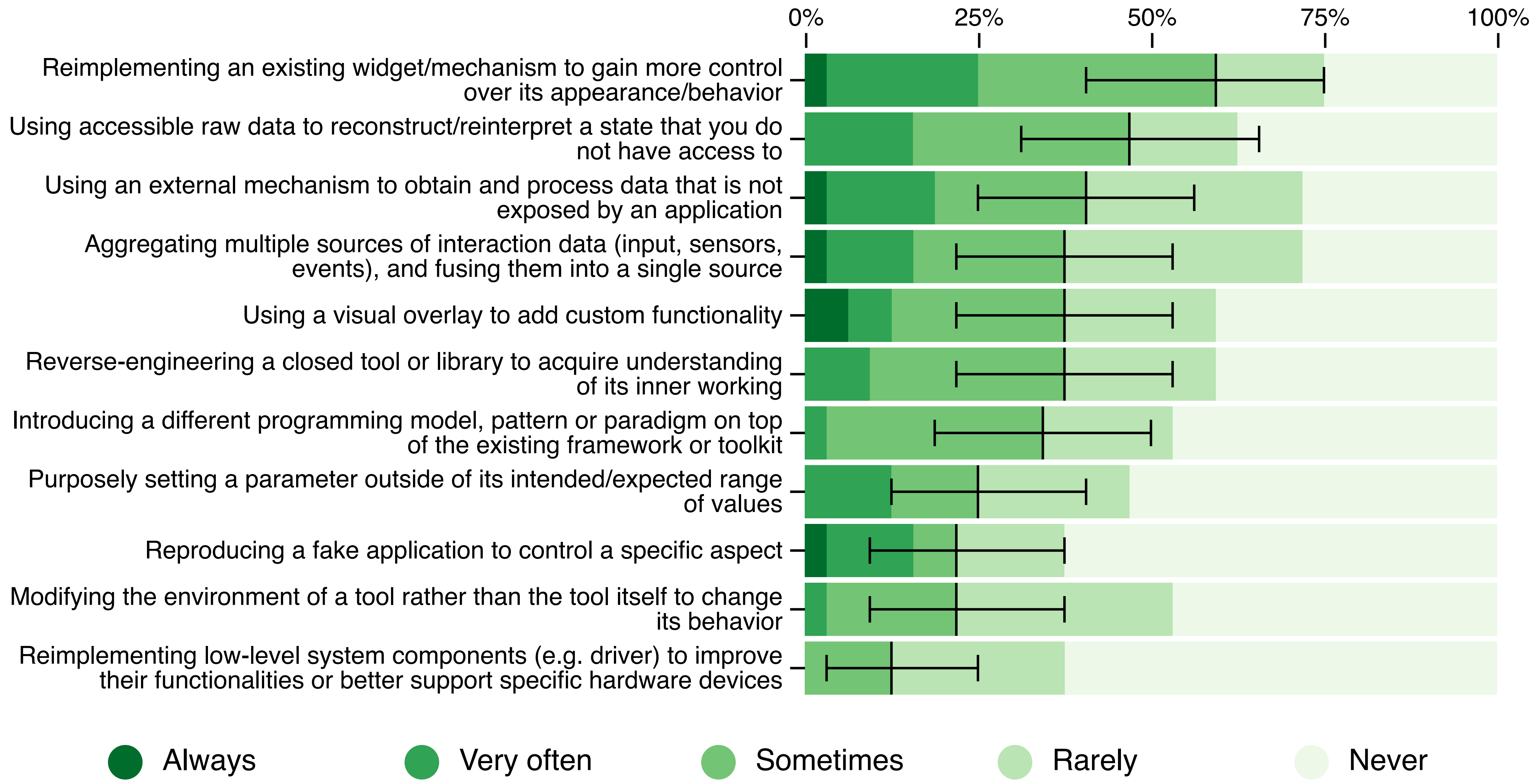
Interviews & Survey

Results





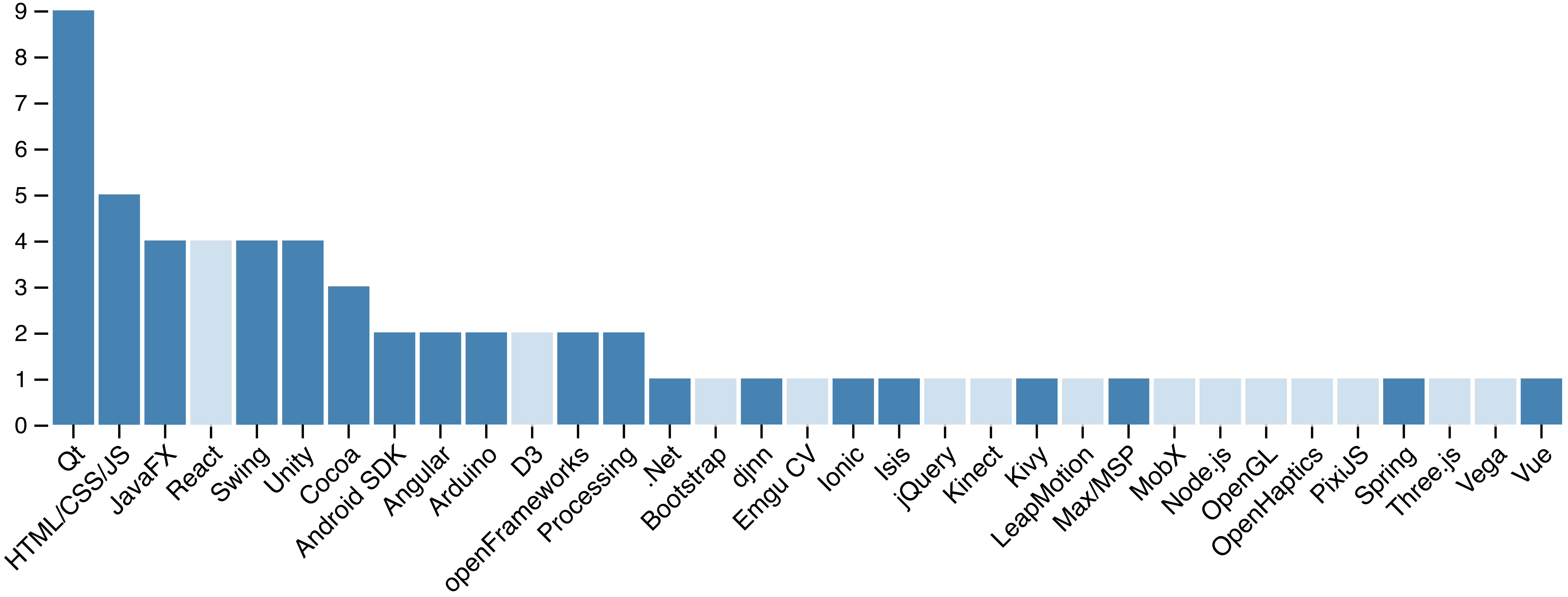




Interviews & Survey

Observation 1

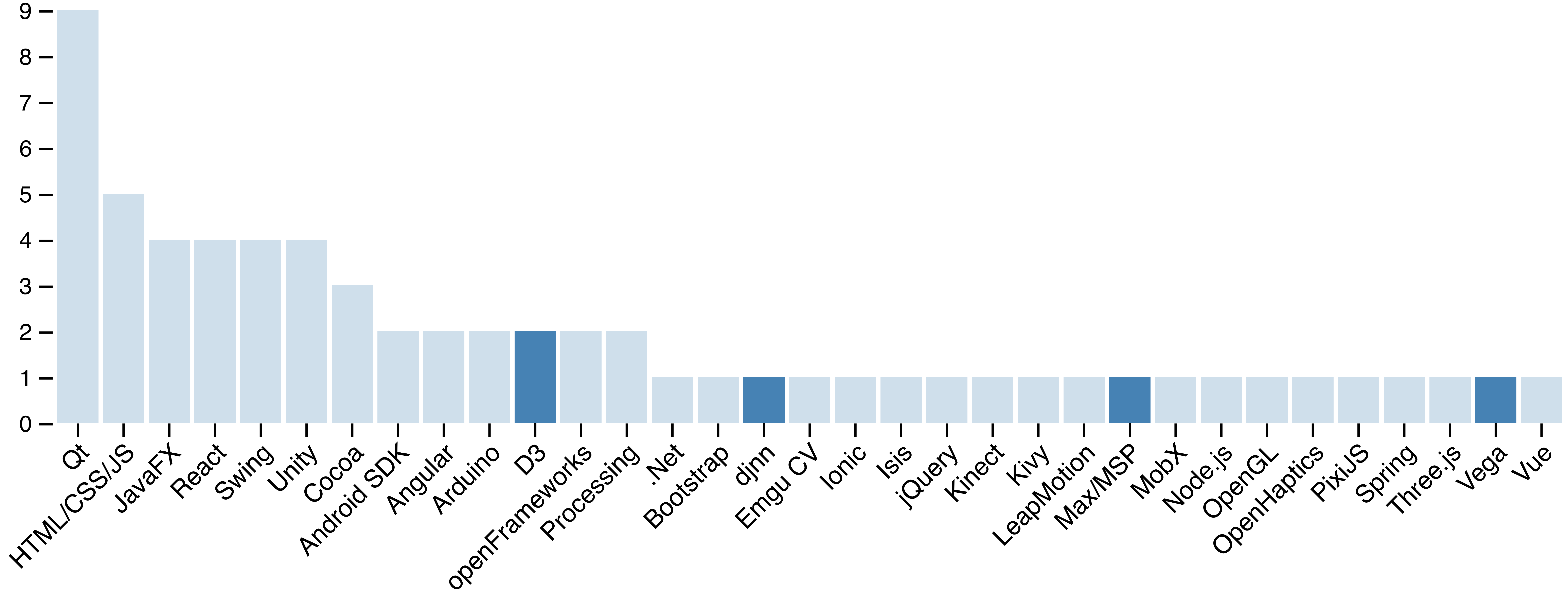
Researchers prioritize well established interaction frameworks over research toolkits



Interviews & Survey

Observation 1

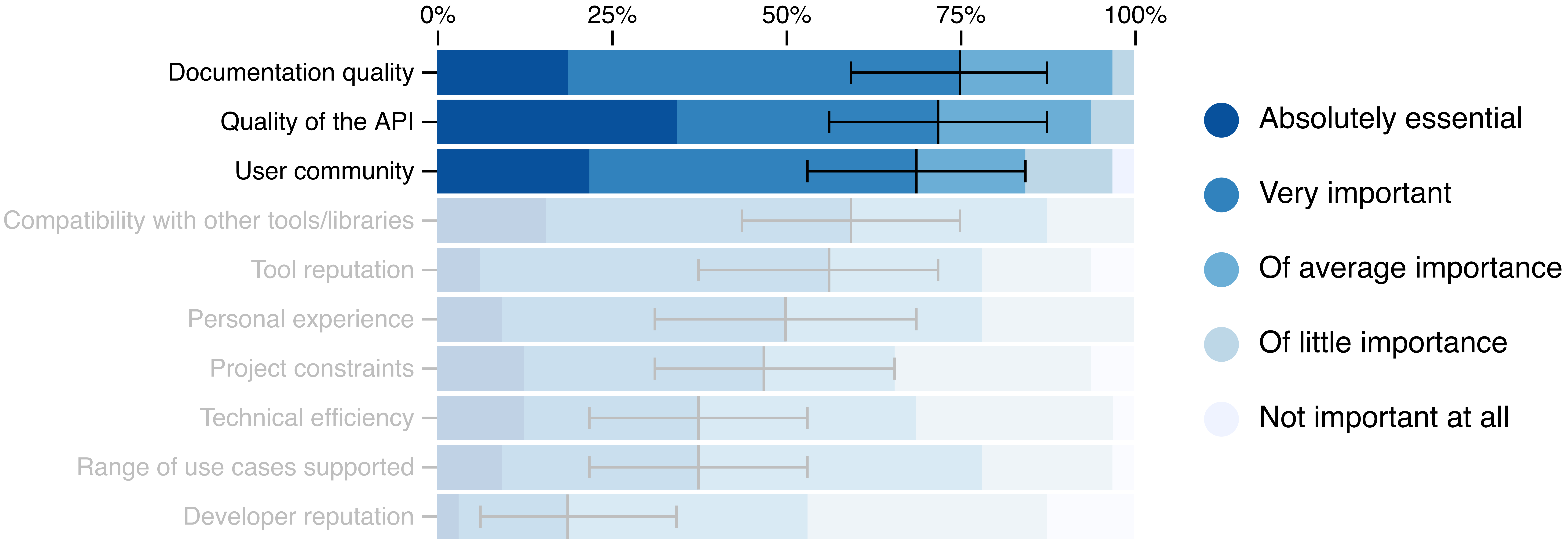
Researchers prioritize well established interaction frameworks over research toolkits



Interviews & Survey

Observation 2

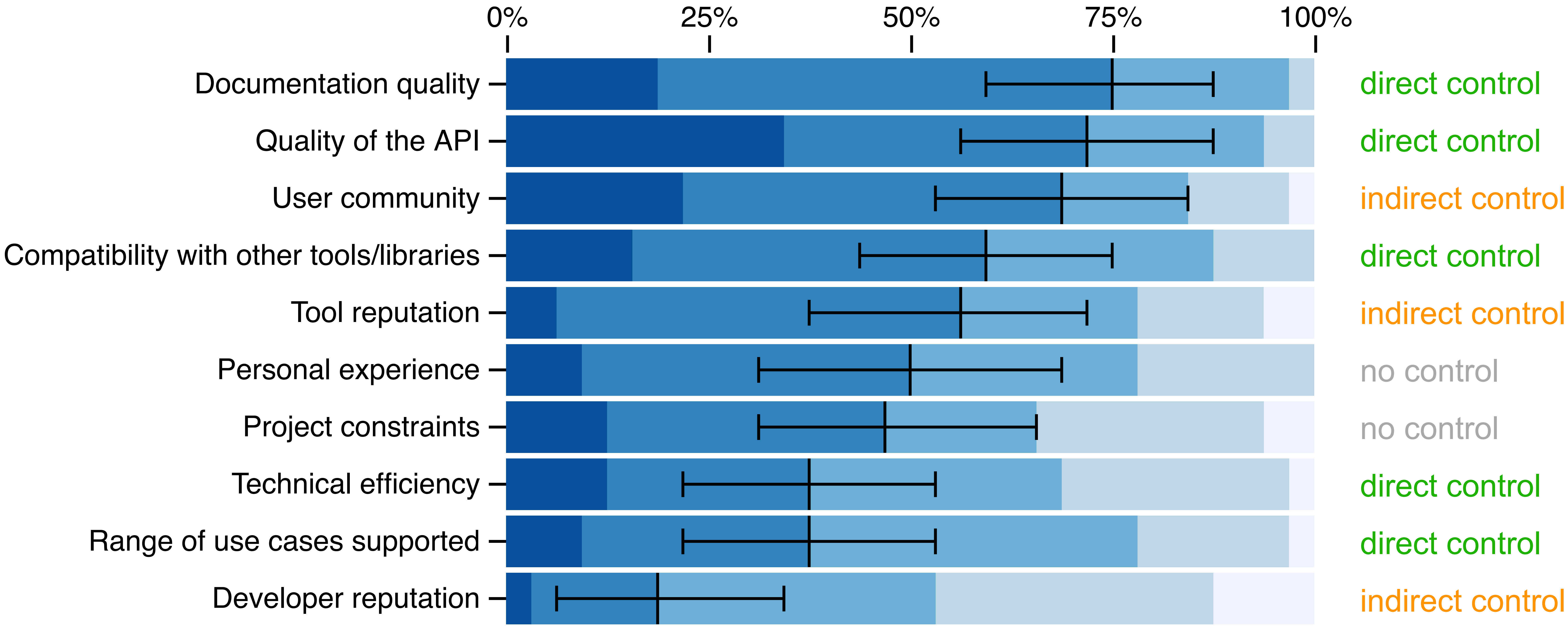
The choice of a library is mostly based on its ease of use, and is directly controlled by its authors



Interviews & Survey

Observation 2

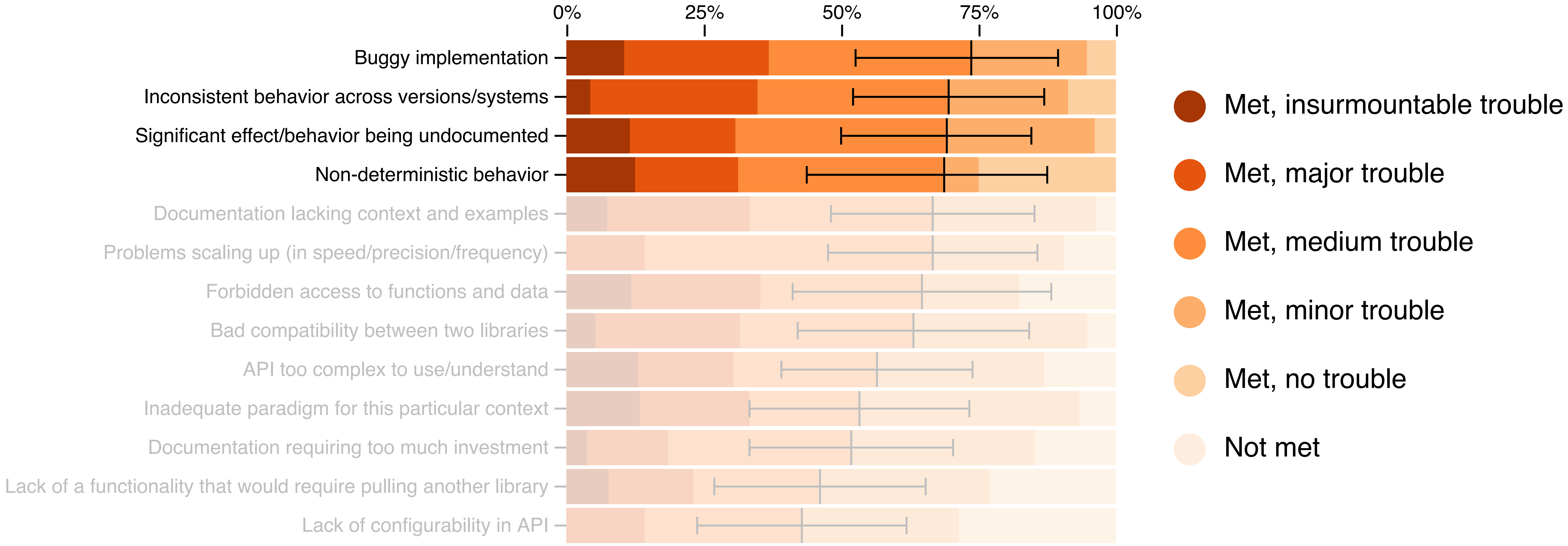
The choice of a library is mostly based on its ease of use, and is directly controlled by its authors



Interviews & Survey

Observation 3

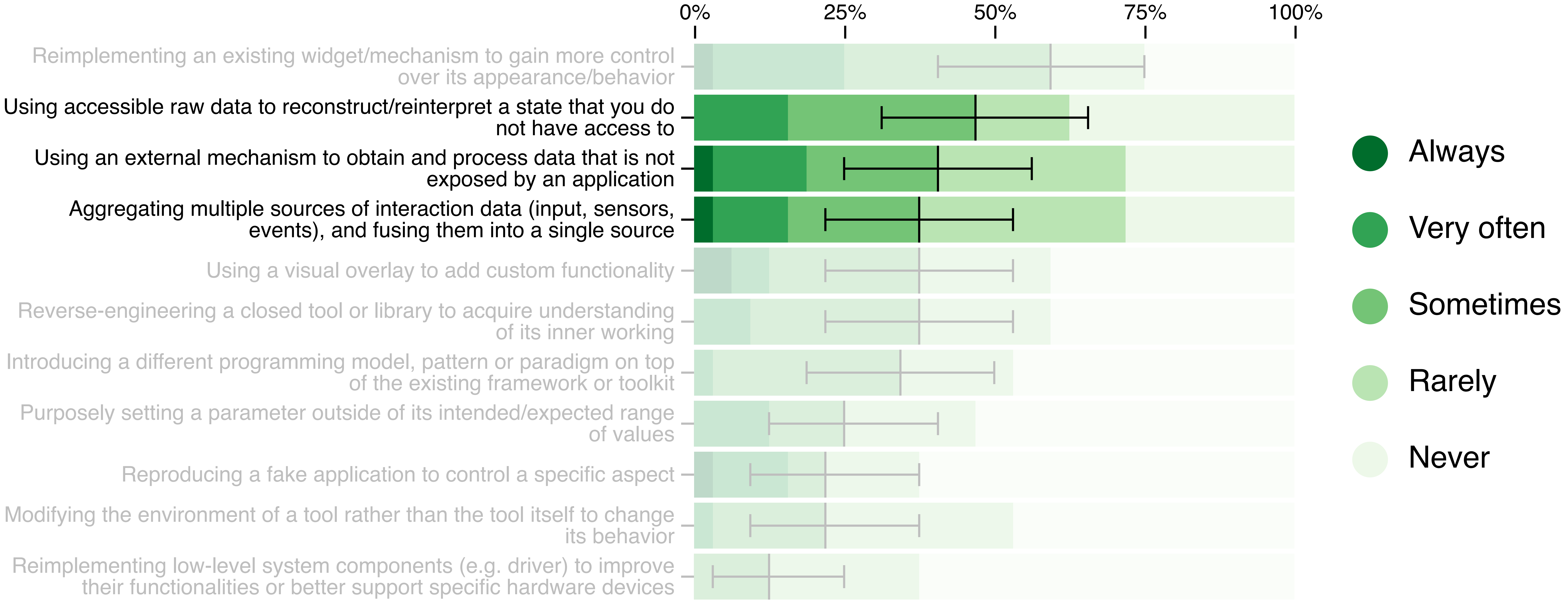
Unpredictability is the most critical problem experienced by researchers with interaction libraries



Interviews & Survey

Observation 4

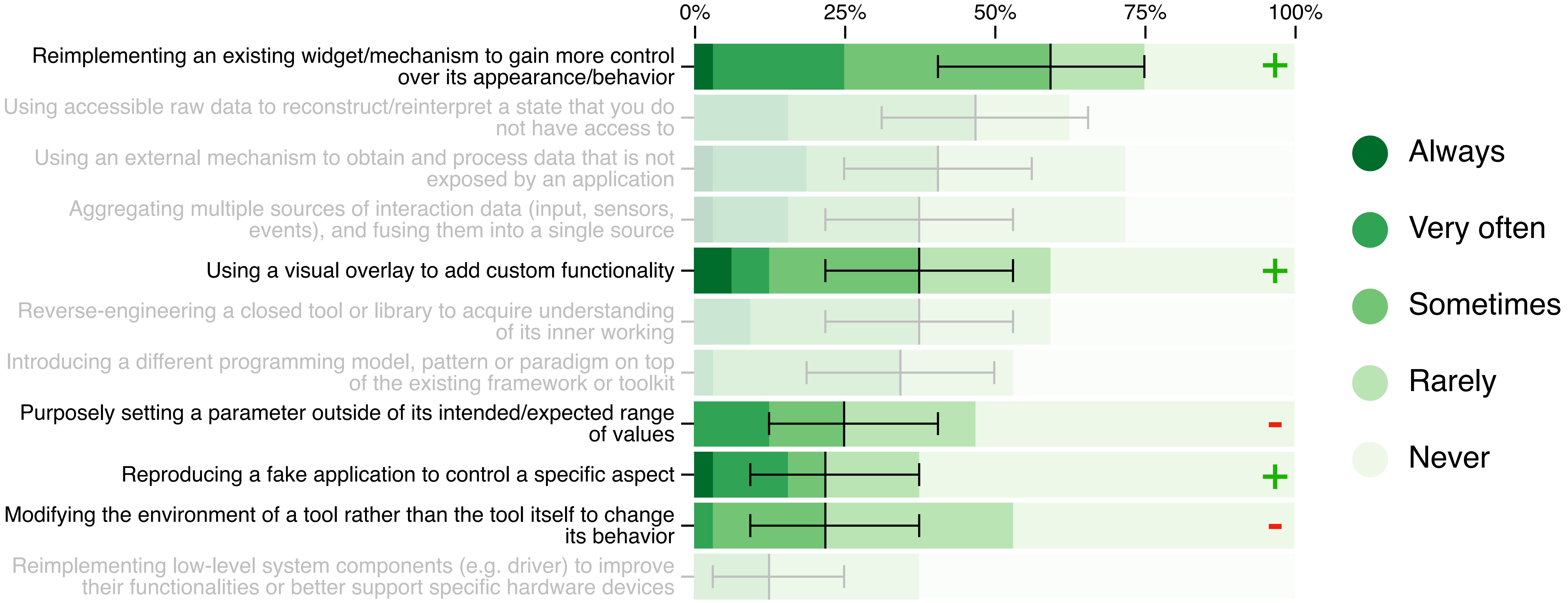
Strategies for gathering and processing interaction data are among the most frequent for our participants



Interviews & Survey

Observation 5

Researchers will often implement new features from scratch rather than patch existing applications or widgets



Interviews & Survey

Takeaways

Obs. 1 → influence frameworks

Obs. 2 → document & test

Obs. 3 → integrate research practices into APIs

Obs. 4 → facilitate access to data

Obs. 5 → promote composition

How can we design or adapt existing frameworks and toolkits to support researchers?

Design recommendations

Related work

Rationales from toolkits:

- rarely discussed in papers
- highly contextual
- lack of justifications on positive impacts

Rationales from frameworks:

- highly abstract
- no general consensus
- lack of tradeoffs acknowledgement

Programming requirements studies:

- good to understand the complexity of frameworks
- need more traction to generate more in-depth descriptions

Design recommendations

Influencing frameworks

How can we have a good impact on frameworks/toolkits?

- code artefact (plugin, toolkit)
- *usage study*
- tech talk (e.g. Qt World Summit, Android Dev Summit)
- join/create a working group
- *design principles*

Duplicate, Accumulate, Defer (DAD)

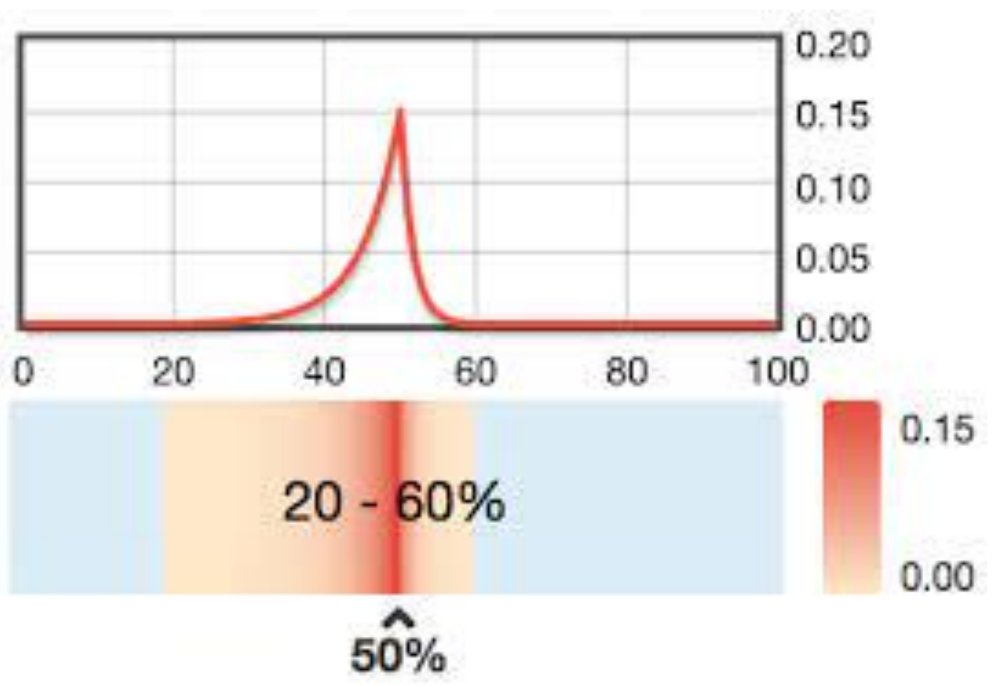
Design recommendations

Duplicate

Allow the duplication of singular elements to foster opportunities for extensions

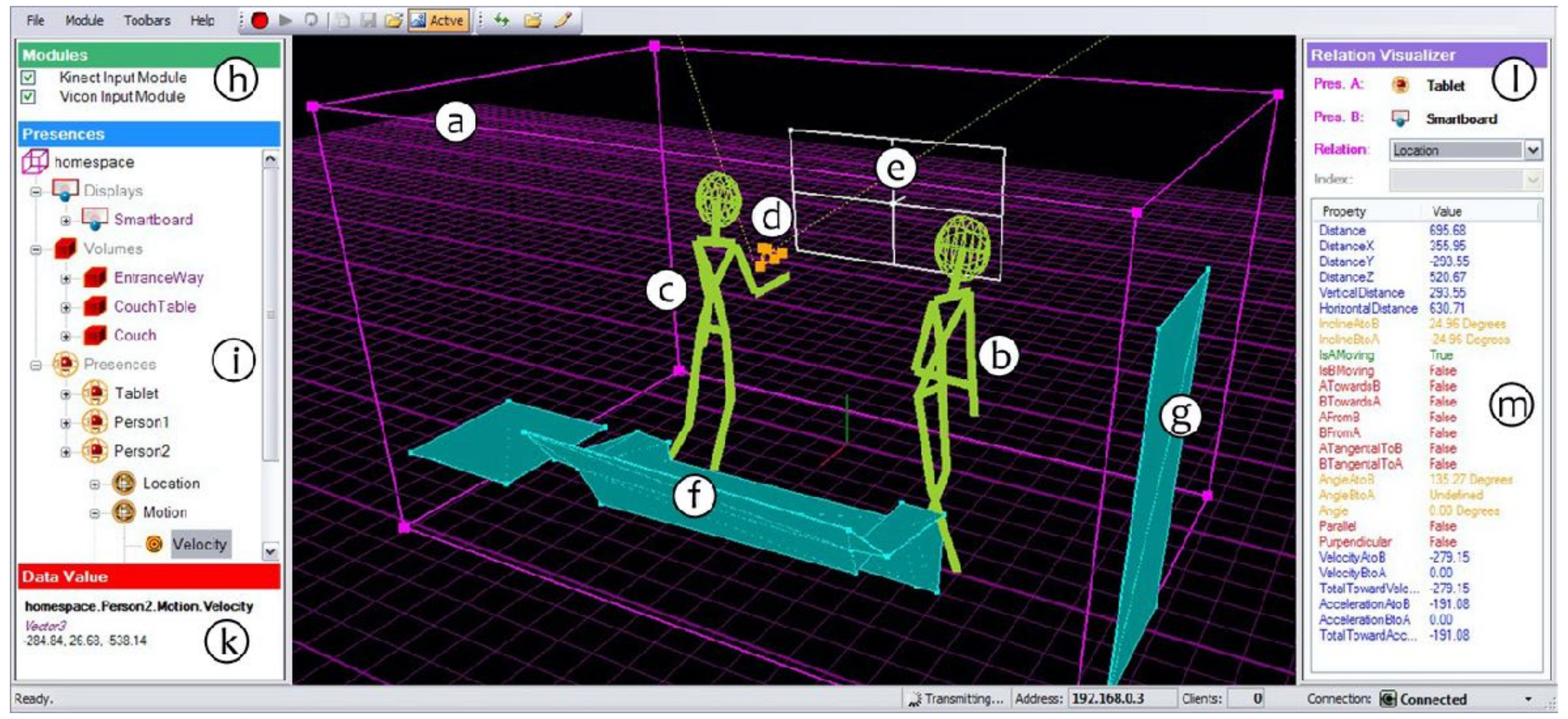
<u>Method</u>: for each element/property/argument

1) Is it expected to be unique?
2) Could it make sense to allow many?



Probability Distribution Sliders (Greis et al.)

ExposeHK (Malacria et al.)



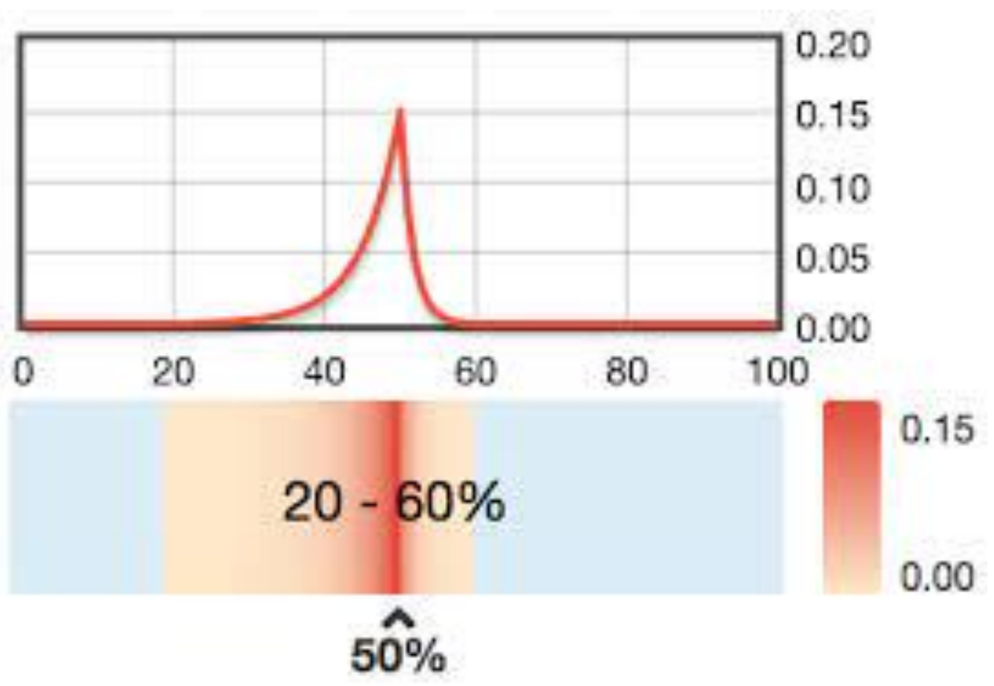
Proximity Toolkit (Marquardt et al.)

Design recommendations

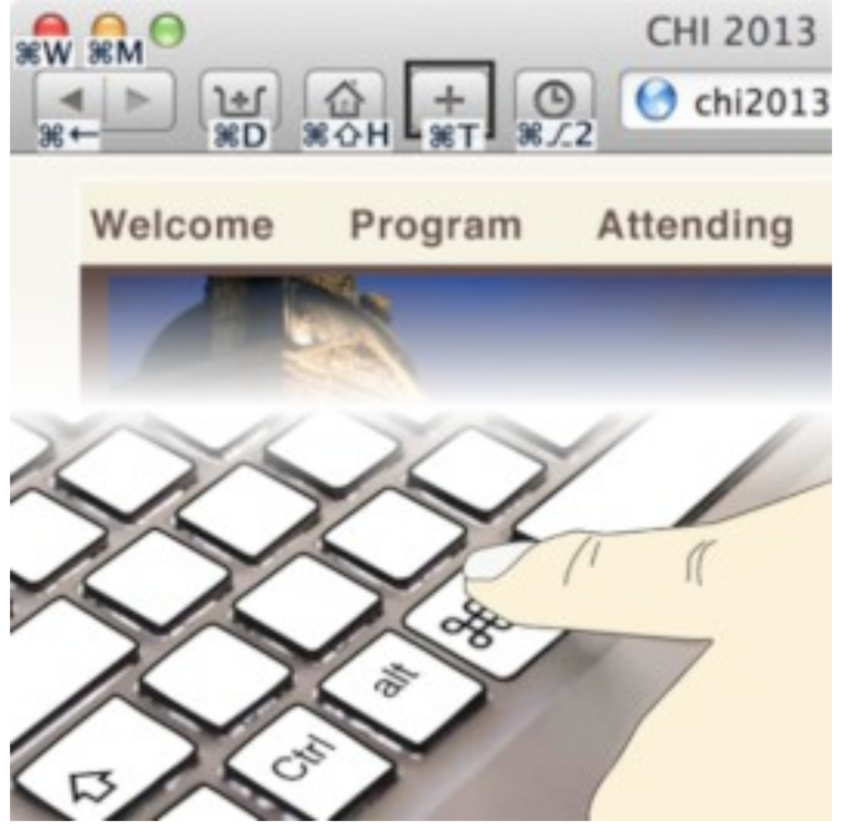
Duplicate

Do not implement these examples → finer reuse/composition

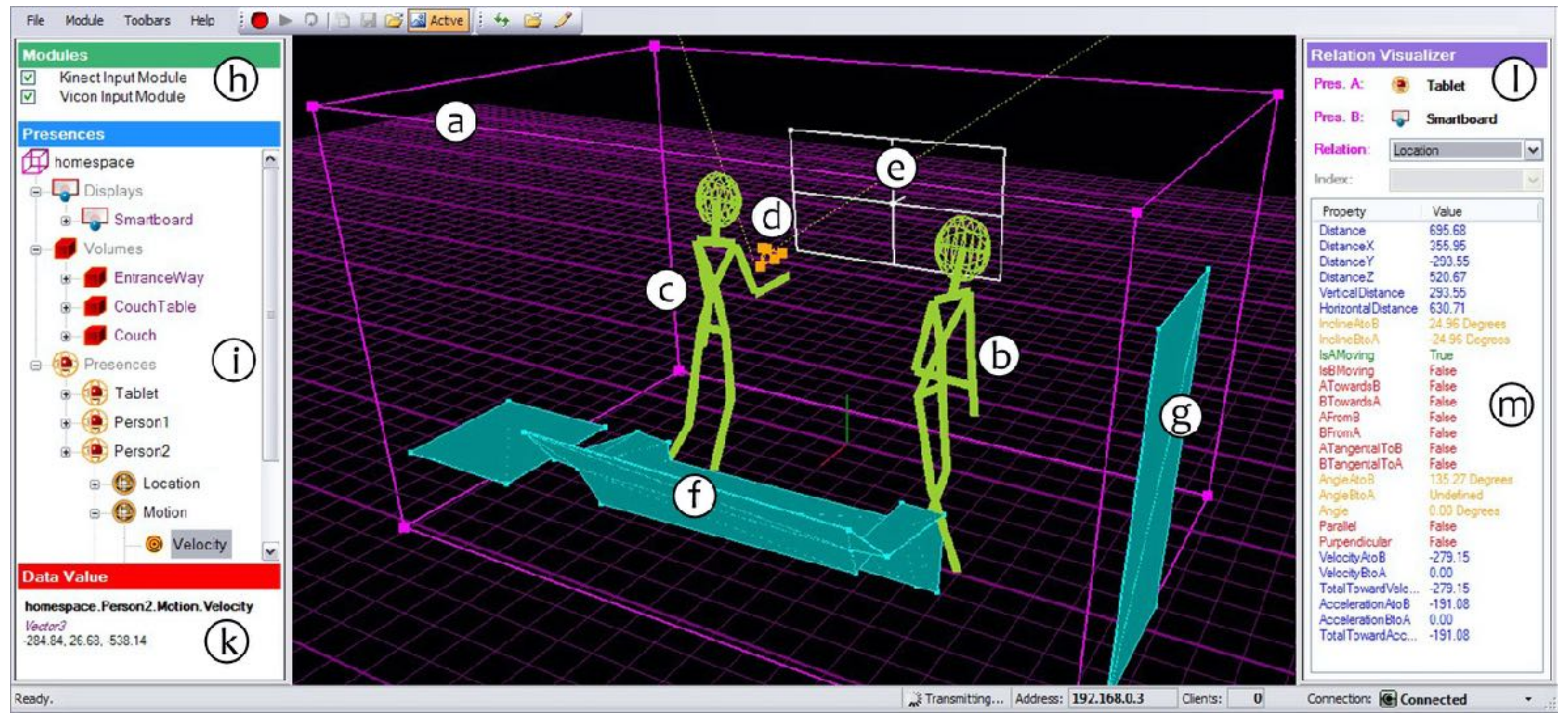
Hard support → toolkits (e.g. multiple mice → libpointing)



Probability Distribution Sliders (Greis et al.)



ExposeHK (Malacria et al.)



Proximity Toolkit (Marquardt et al.)

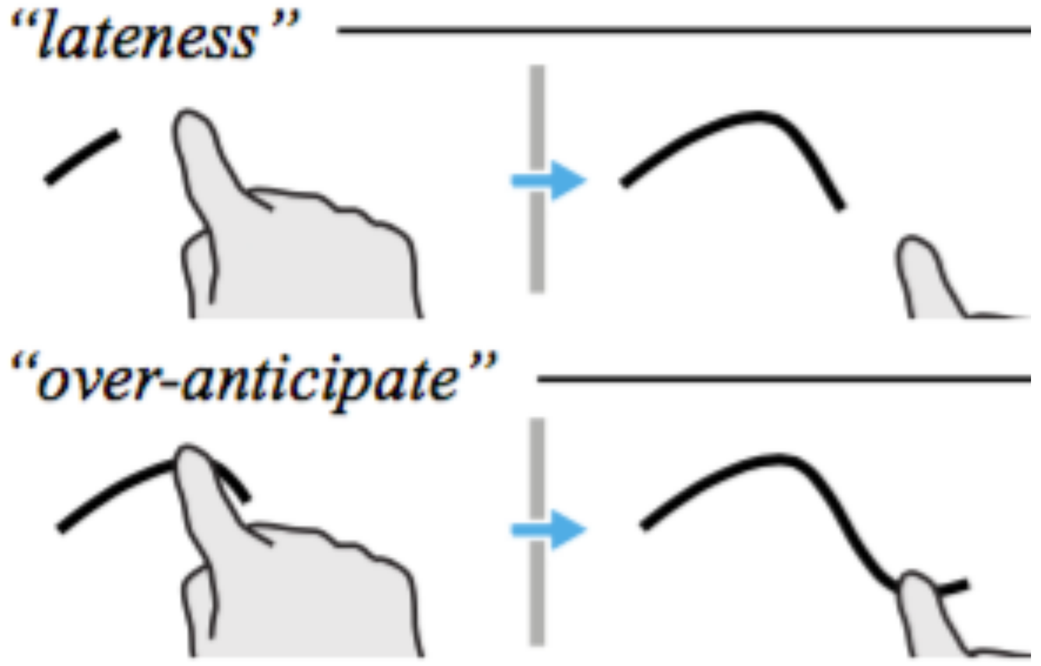
Design recommendations

Accumulate

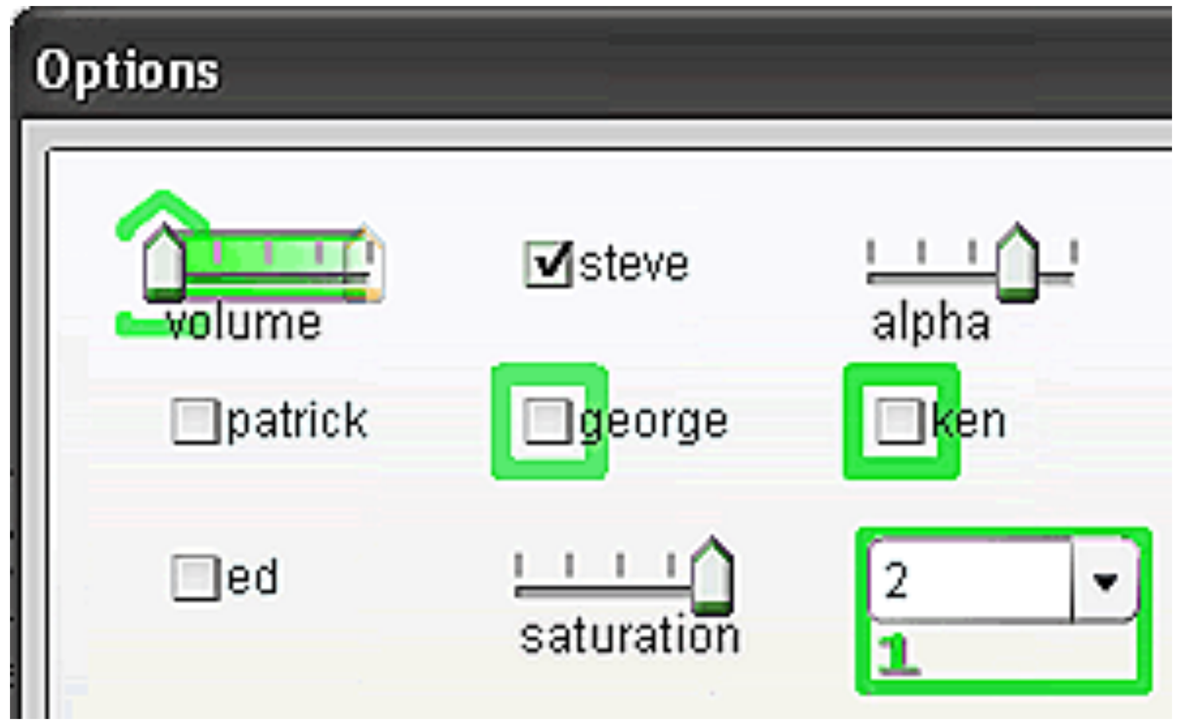
Accumulate rather than replace to keep a history of changes

Method: for each property/argument

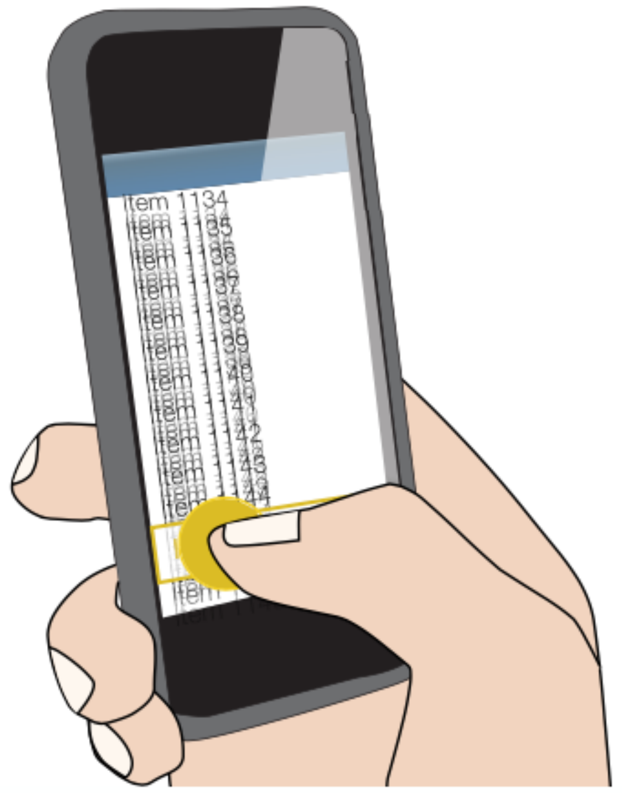
1) Is this data replaced by another?
2) Could it make sense to keep both at any time?



TurboTouch (Nancel et al.)



Phosphor (Baudisch et al.)



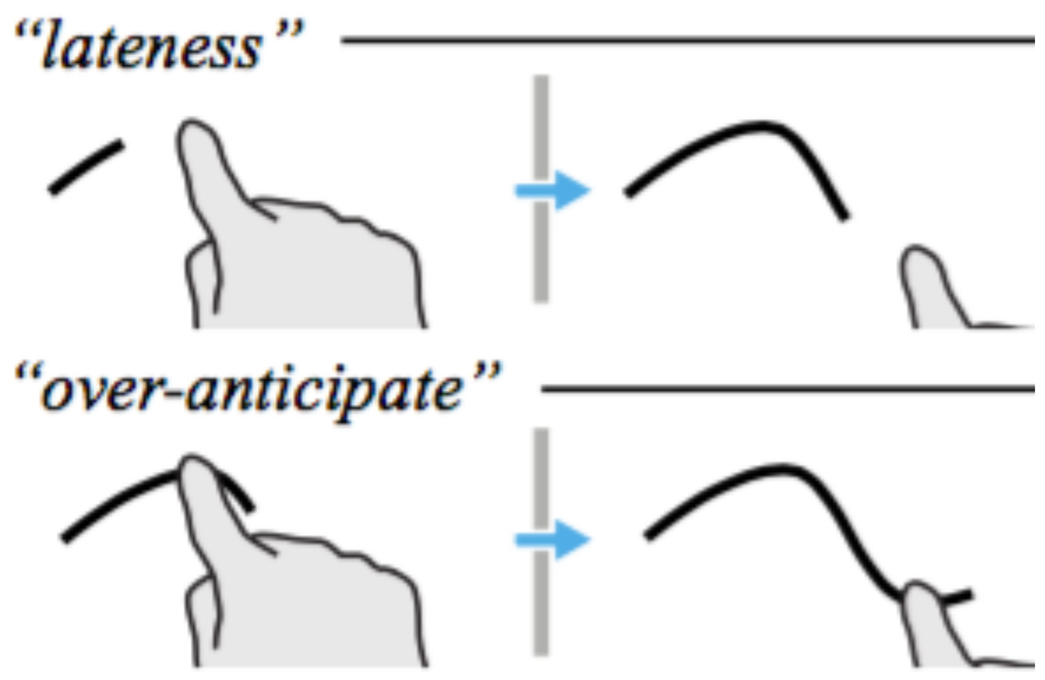
ForceEdge (Antoine et al.)

Design recommendations

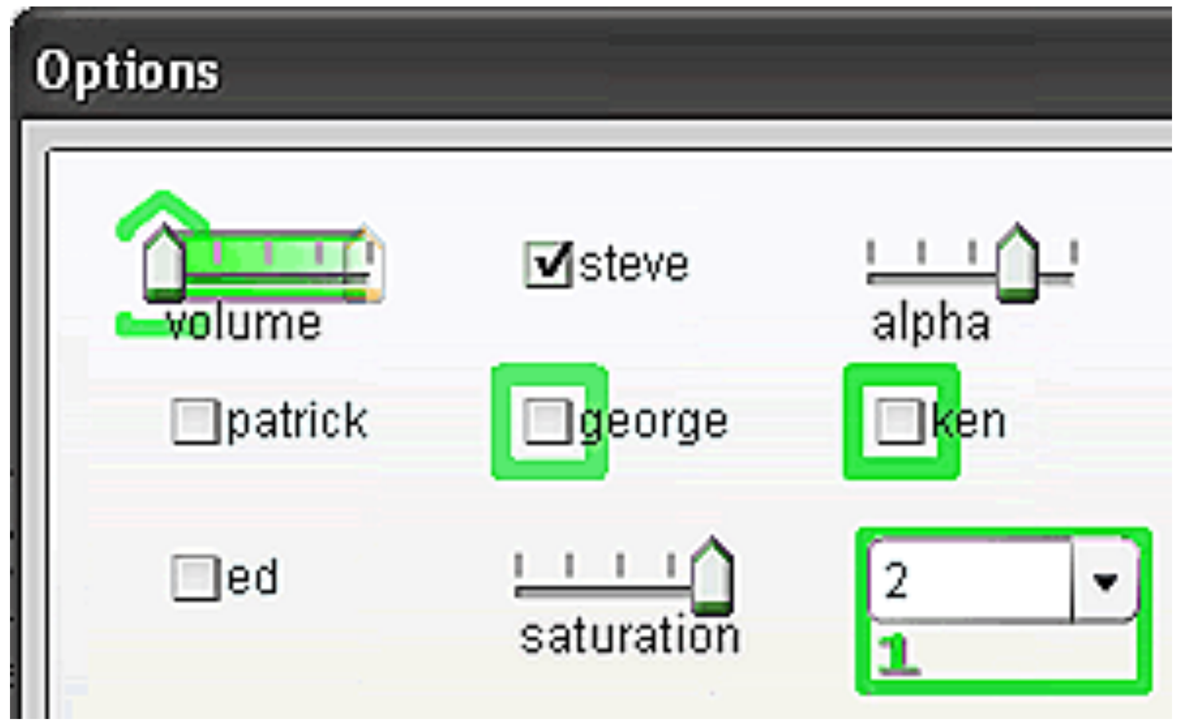
Accumulate

Accumulation over time/space

Polymorphism



TurboTouch (Nancel et al.)



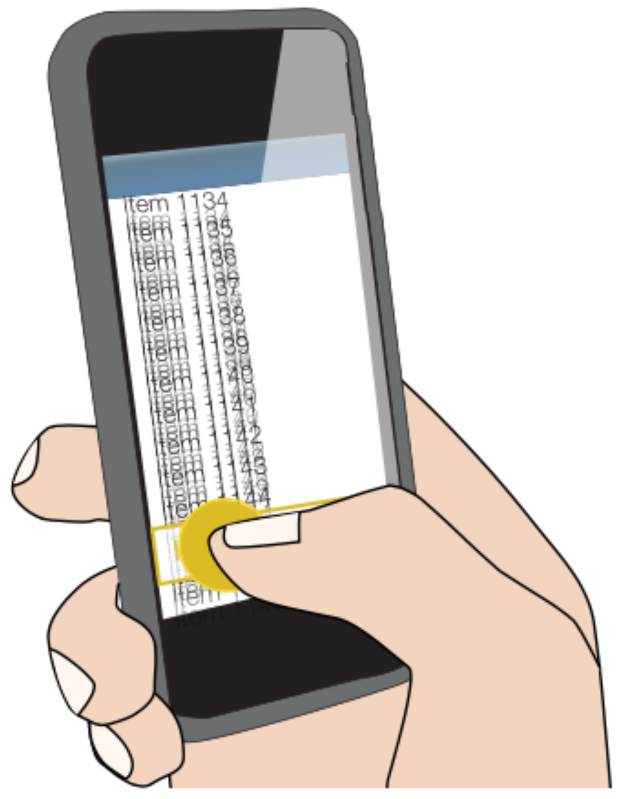
Options

volume ☑steve alpha

□patrick □george □ken

□ed saturation 2 1

Phosphor (Baudisch et al.)



ForceEdge (Antoine et al.)

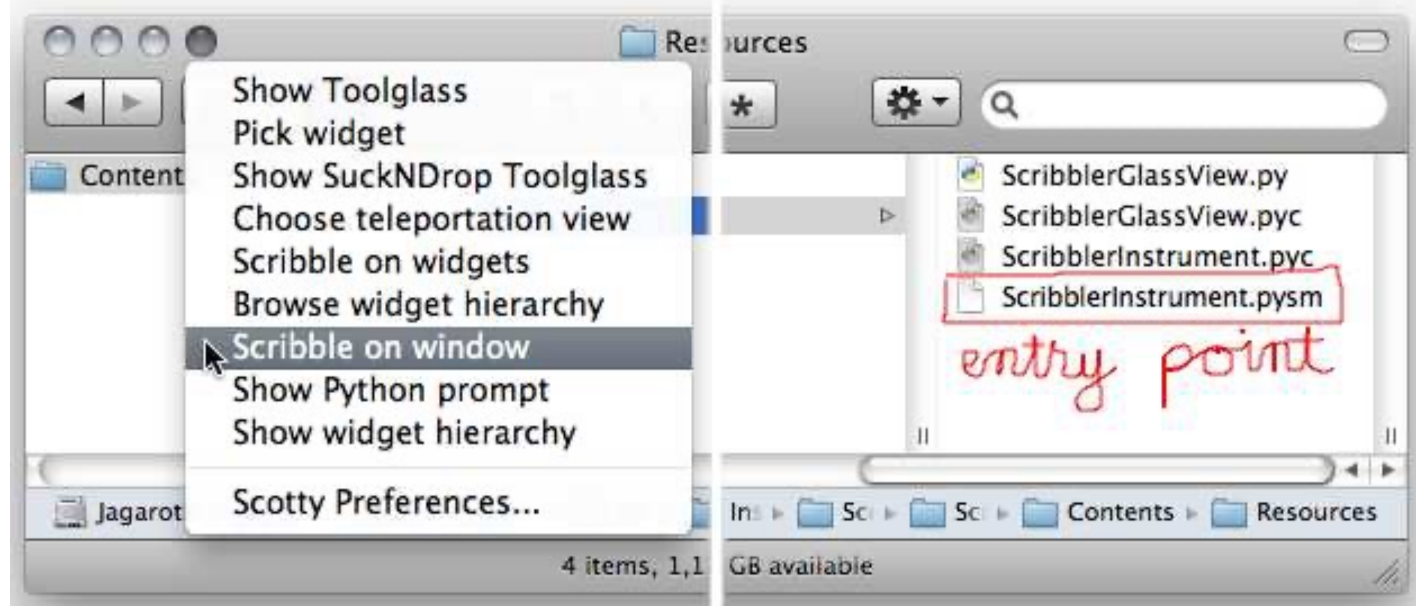
Design recommendations

Defer

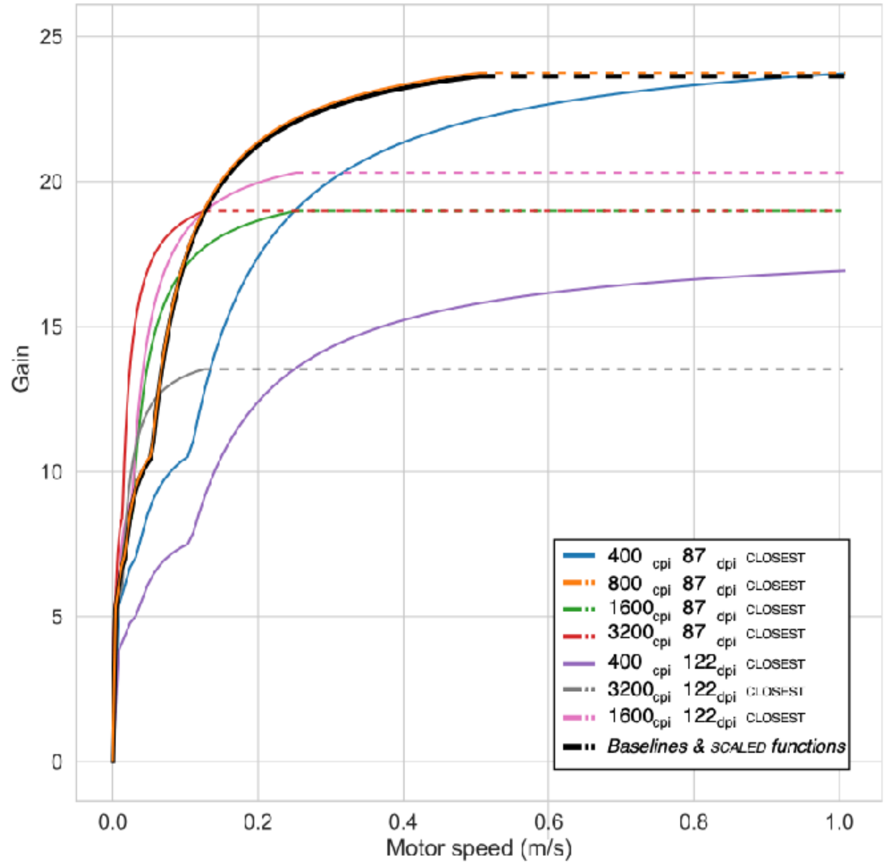
Defer the execution of predefined behaviors to enable their monitoring and replacement

Method: for each function/method

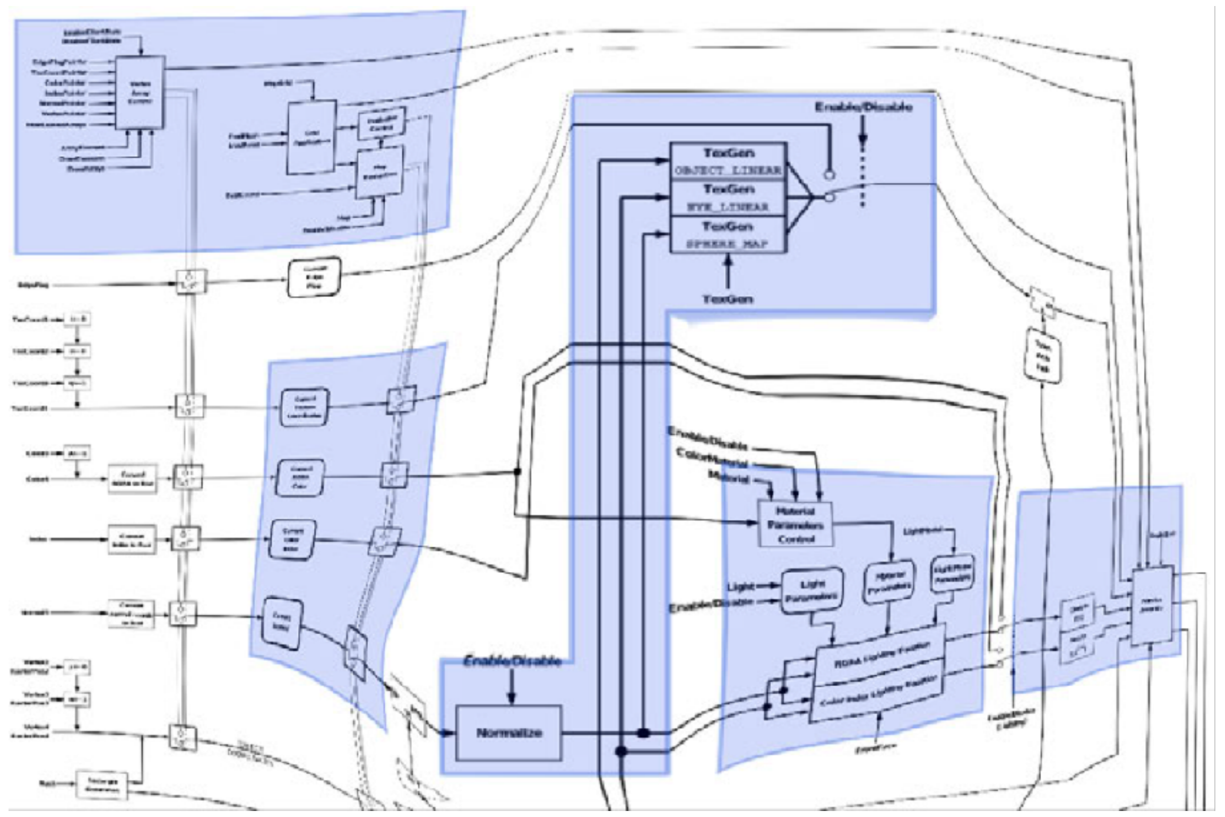
1) Can this action be intercepted? (i.e. canceled, altered or repeated)
2) If not, could it be useful at run-time or compile-time?



Scotty (Eagan et al.)



libpointing (Casiez et al.)



JellyLens (Pindat et al.)

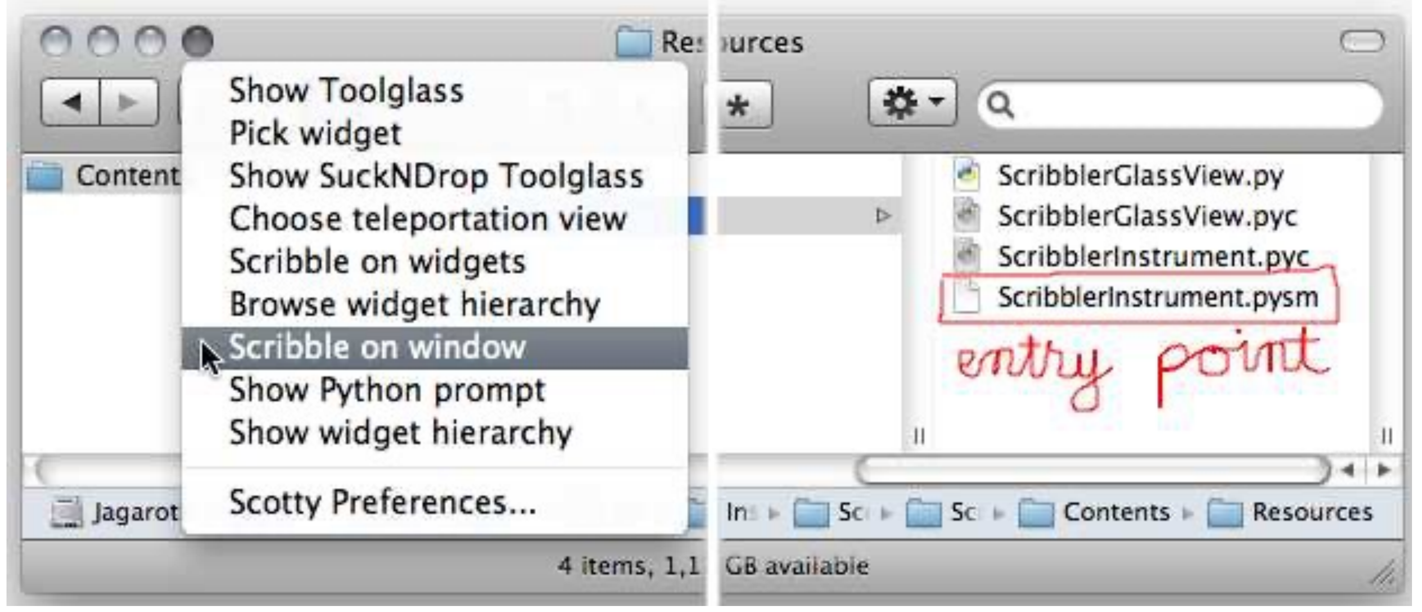
Design recommendations

Defer

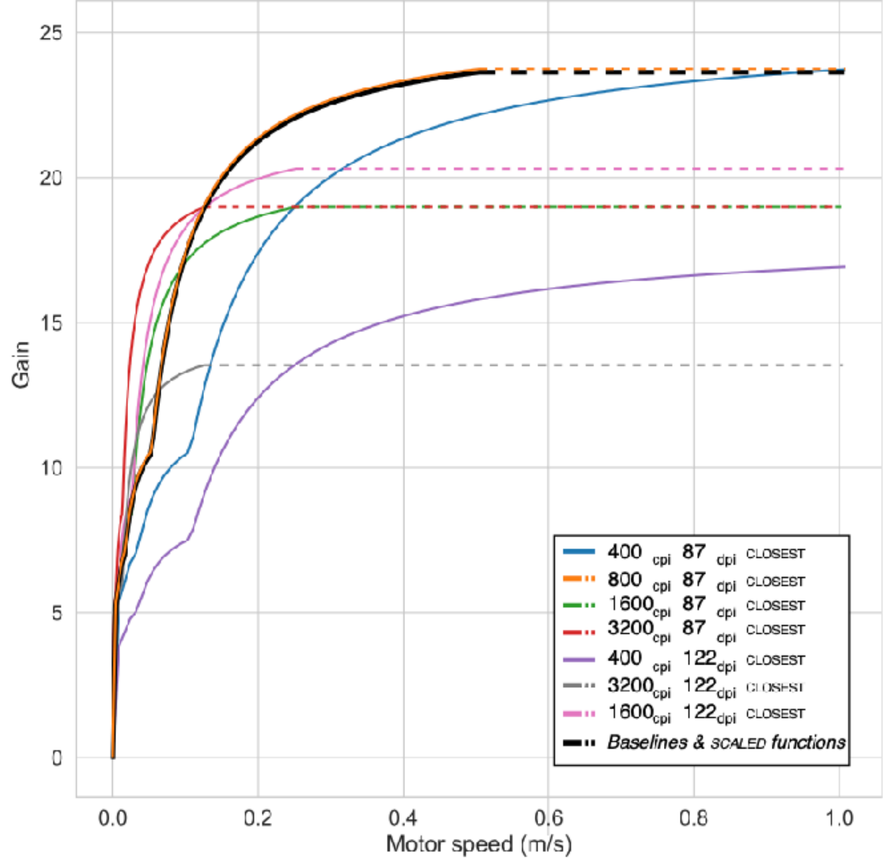
Split commands into (i) placing an order and (ii) executing it

More scalable indirection mechanisms:

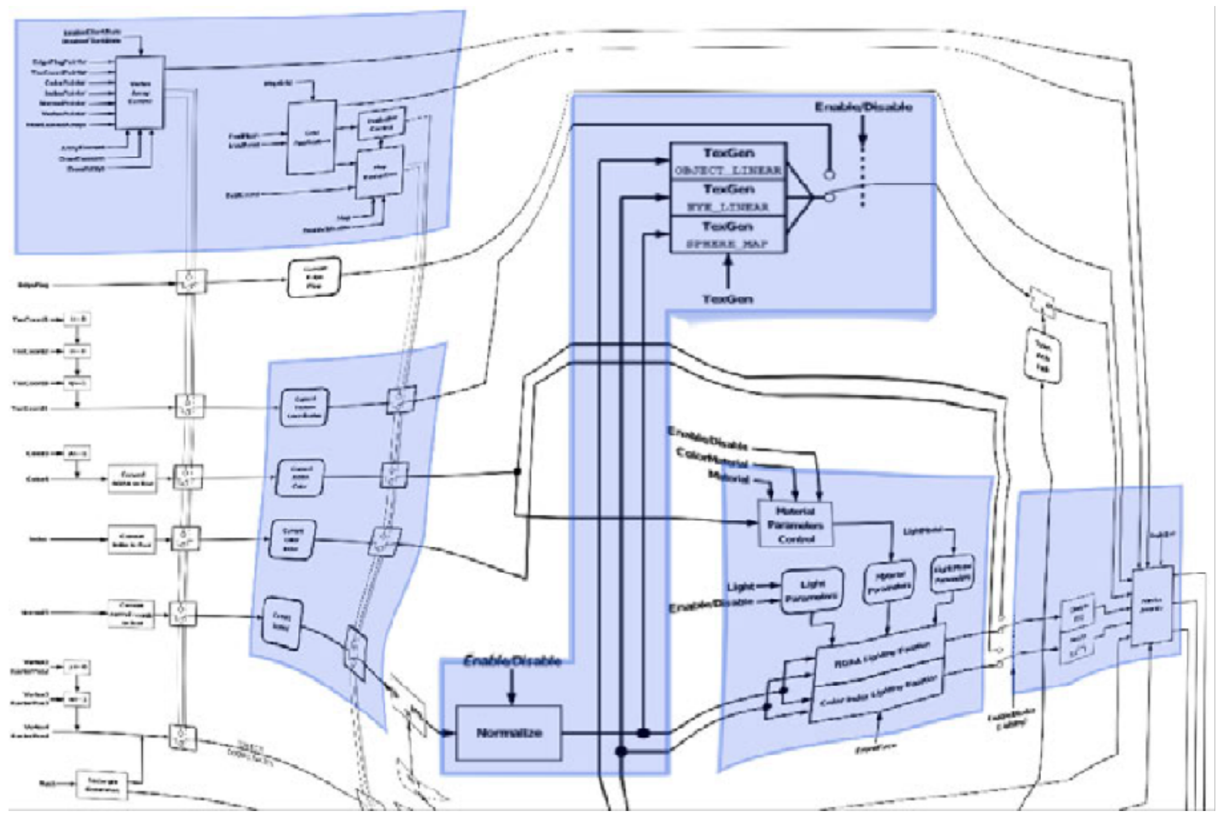
- open intermediate structures (e.g. DOM, framebuffer)
- software buses



Scotty (Eagan et al.)



libpointing (Casiez et al.)



JellyLens (Pindat et al.)

Conclusion and future work

Contributions:

- key observations about researchers when programming novel interaction techniques
- design principles to better support them in frameworks & toolkits

Future work:

- promoting these principles
- classifying programming practices vs types of interaction techniques
- evaluating how much the principles are applied already

Thank you for your attention