What do Researchers Need when Implementing Novel Interaction Techniques?

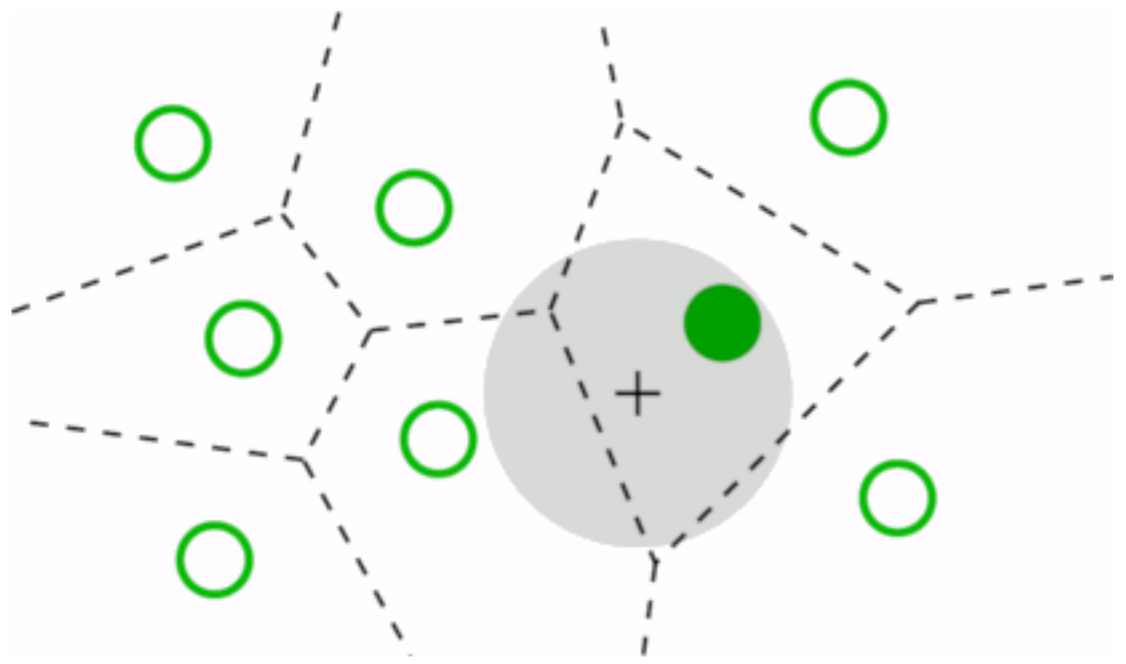
Thibault Raffaillac
École Centrale de Lyon

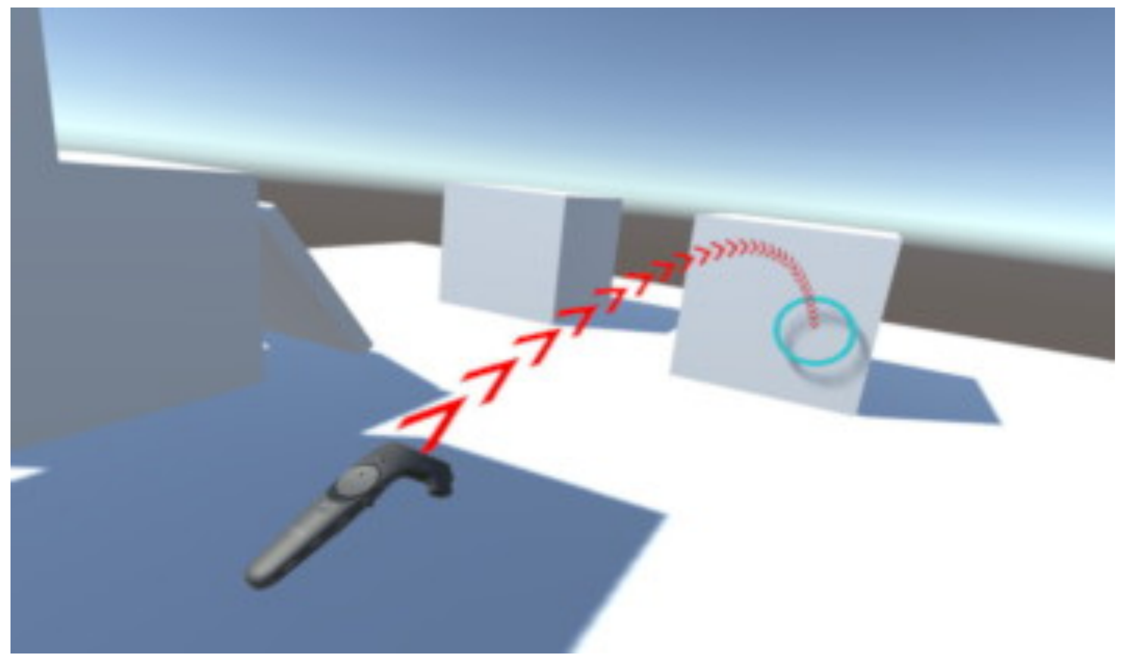
Stéphane Huot
Inria Lille – Nord Europe

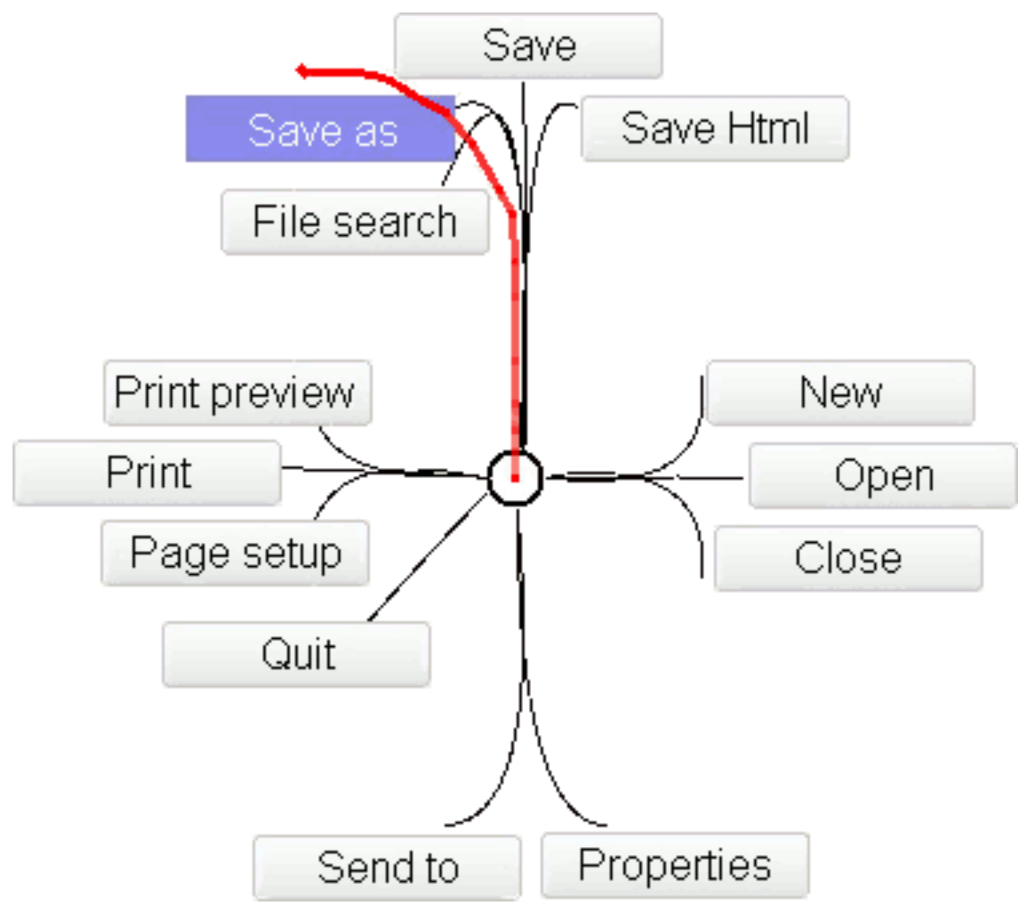
Introduction

Motivation

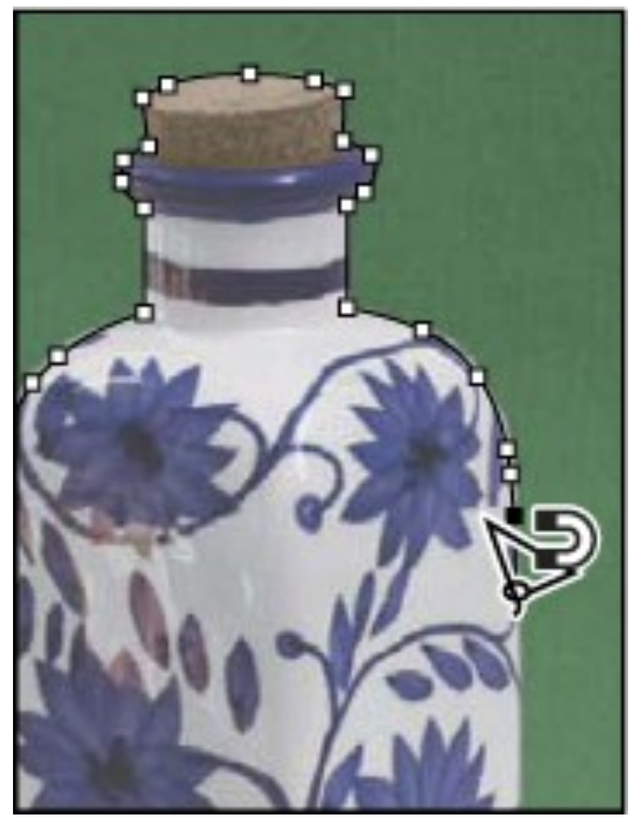
Frustration of colleagues when programming novel interaction techniques for research


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

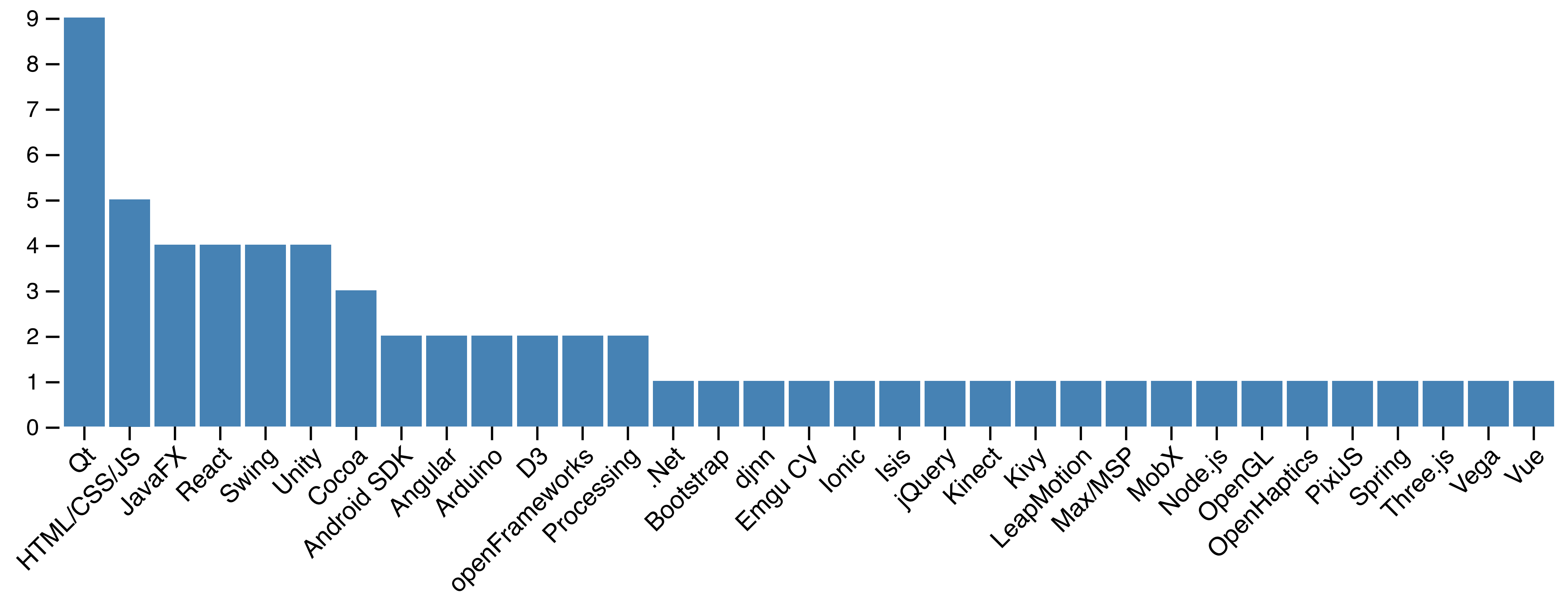
32 survey participants
CHI community
2/3 advanced or experts
Rating predefined items

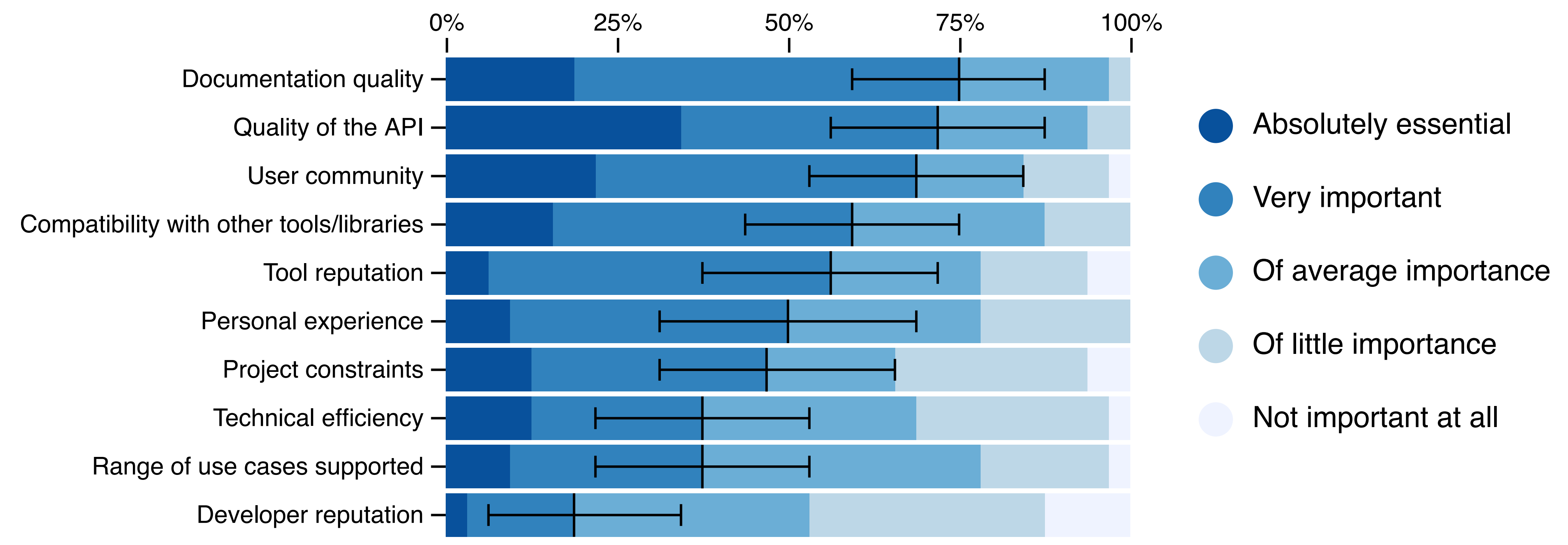
Quantitative analysis

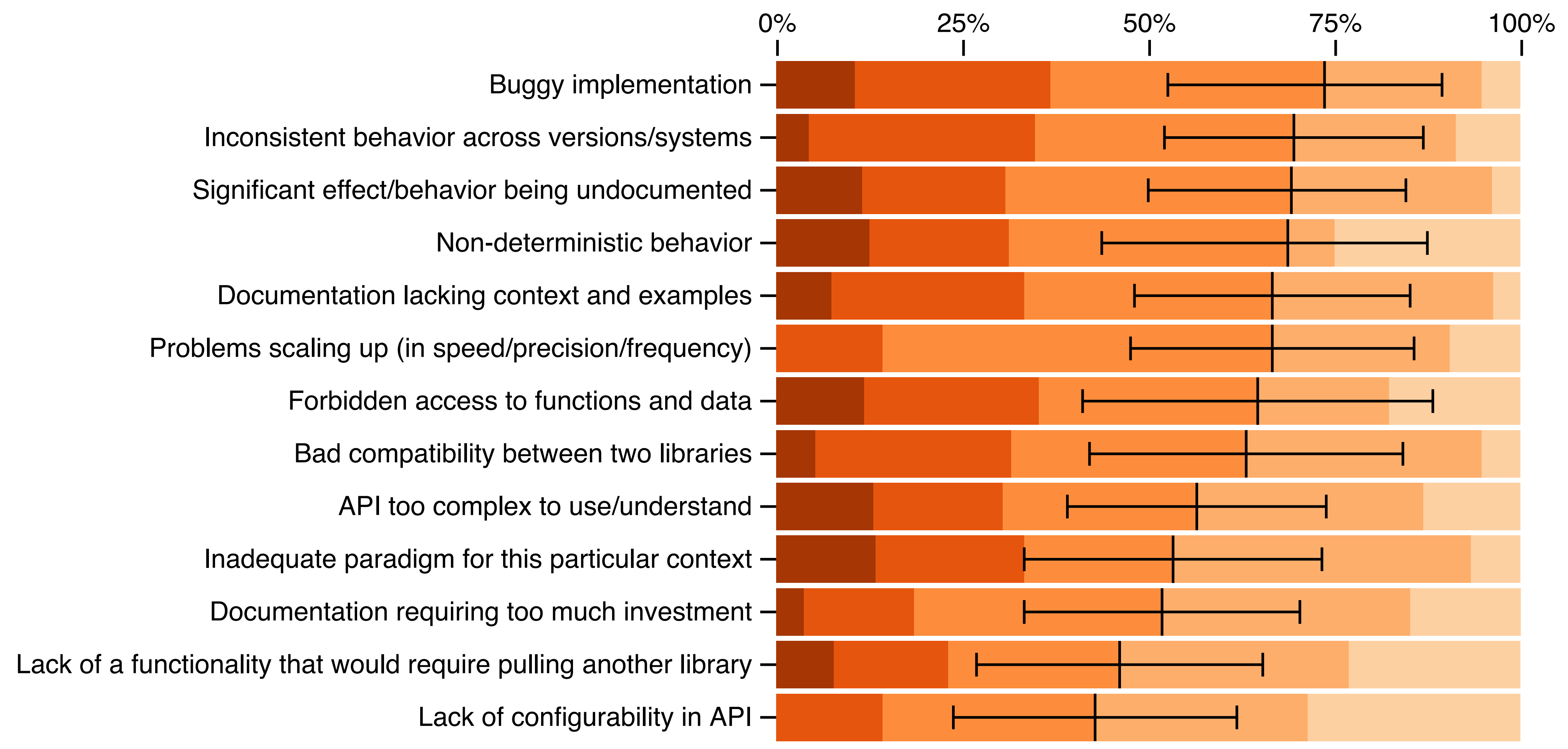
3 rankings:
- criteria of choice (R1)
- severity of problems (R2)
- frequency of strategies (R3)

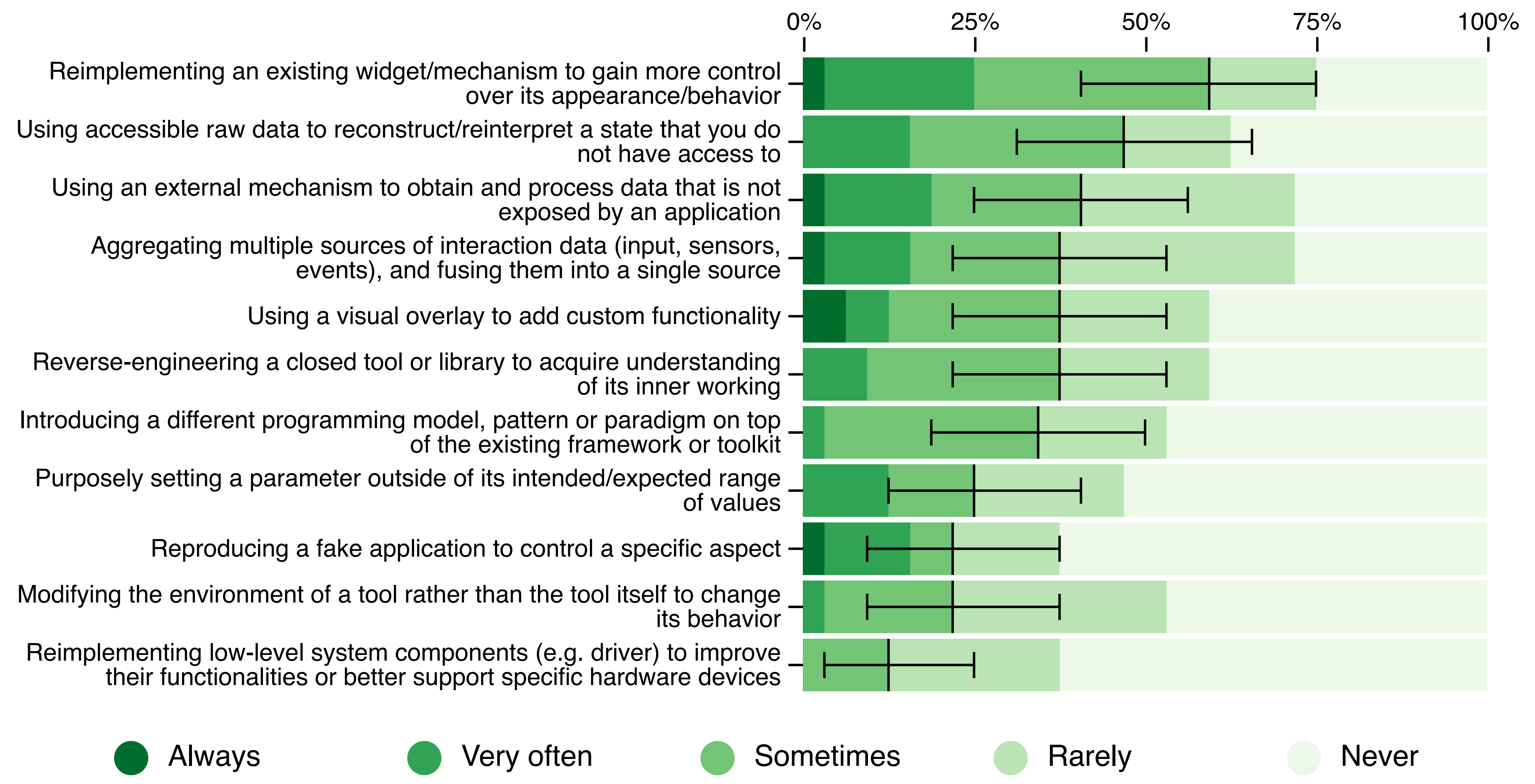
Interviews & Survey

Results





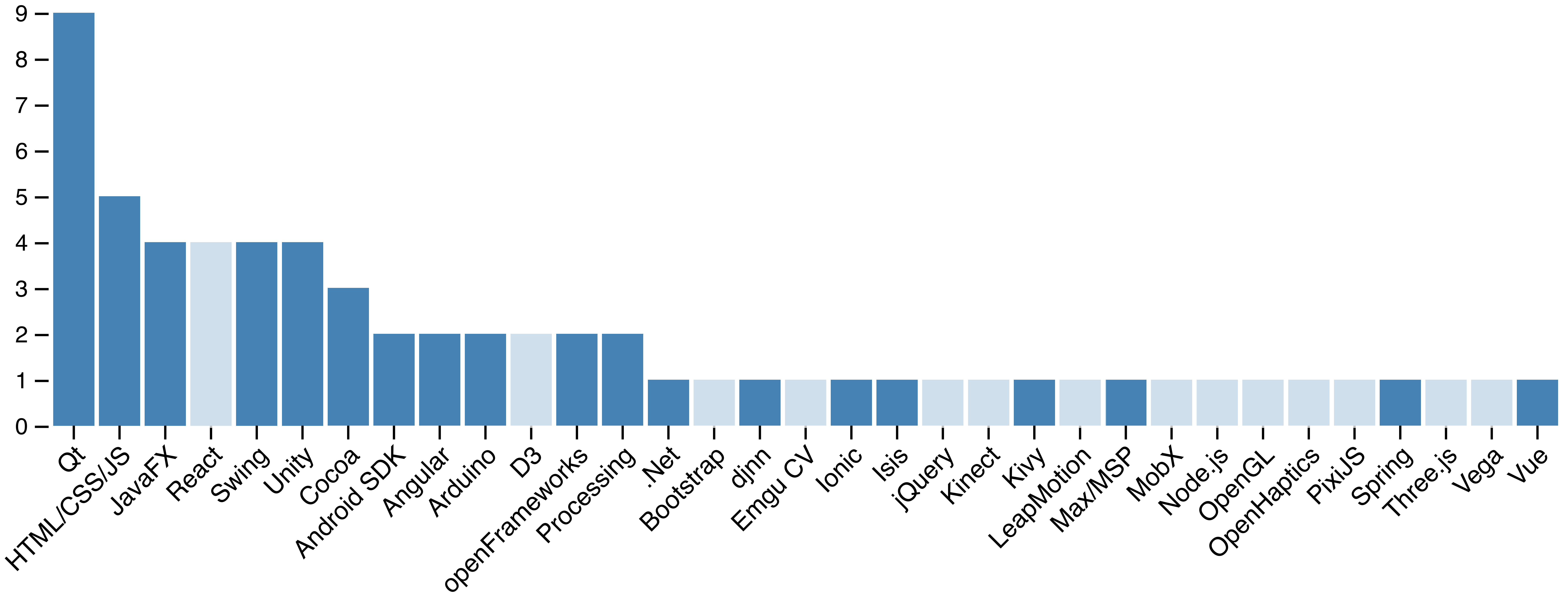




Interviews & Survey

Observation 1

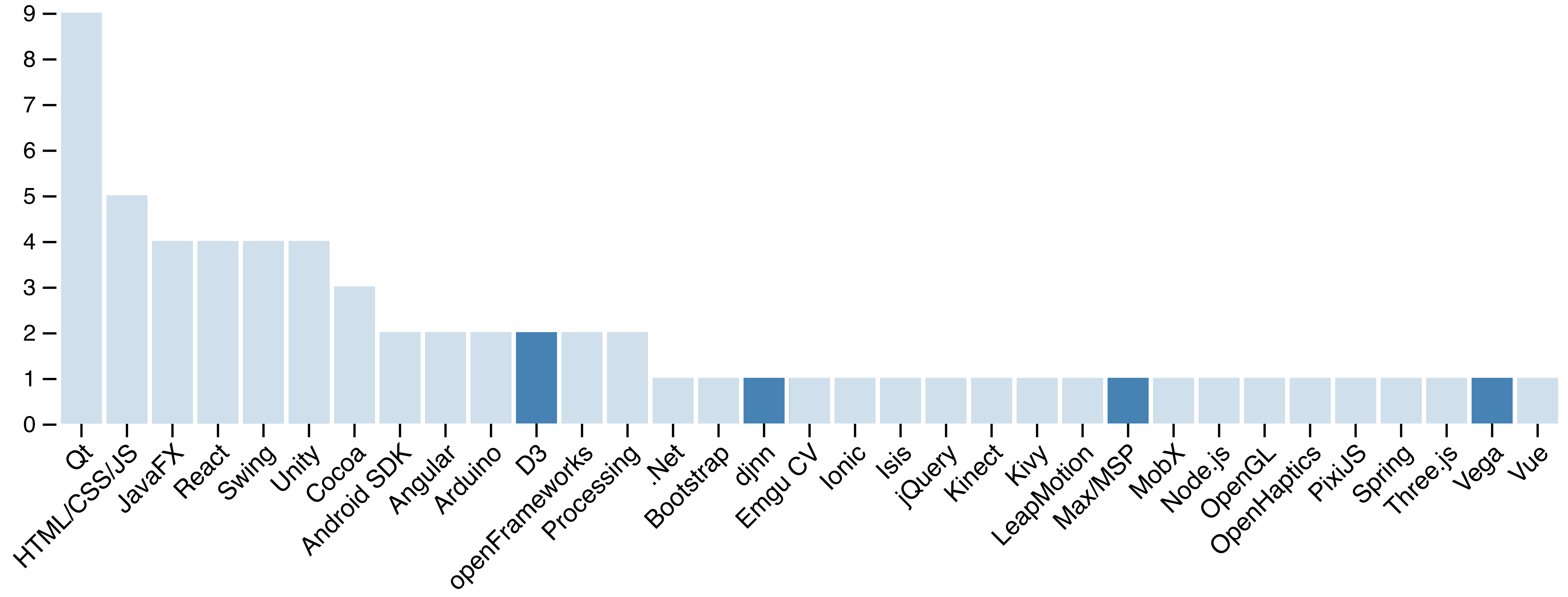
Researchers prioritize well established interaction frameworks over research toolkits



Interviews & Survey

Observation 1

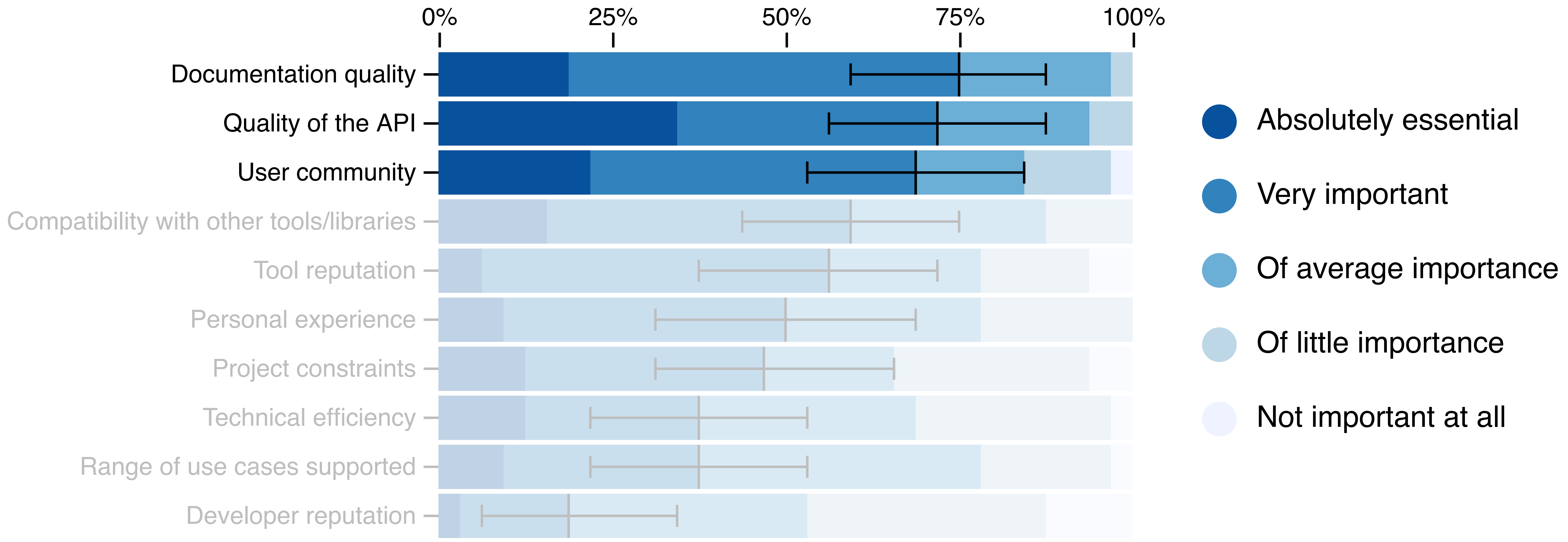
Researchers prioritize well established interaction frameworks over research toolkits



Interviews & Survey

Observation 2

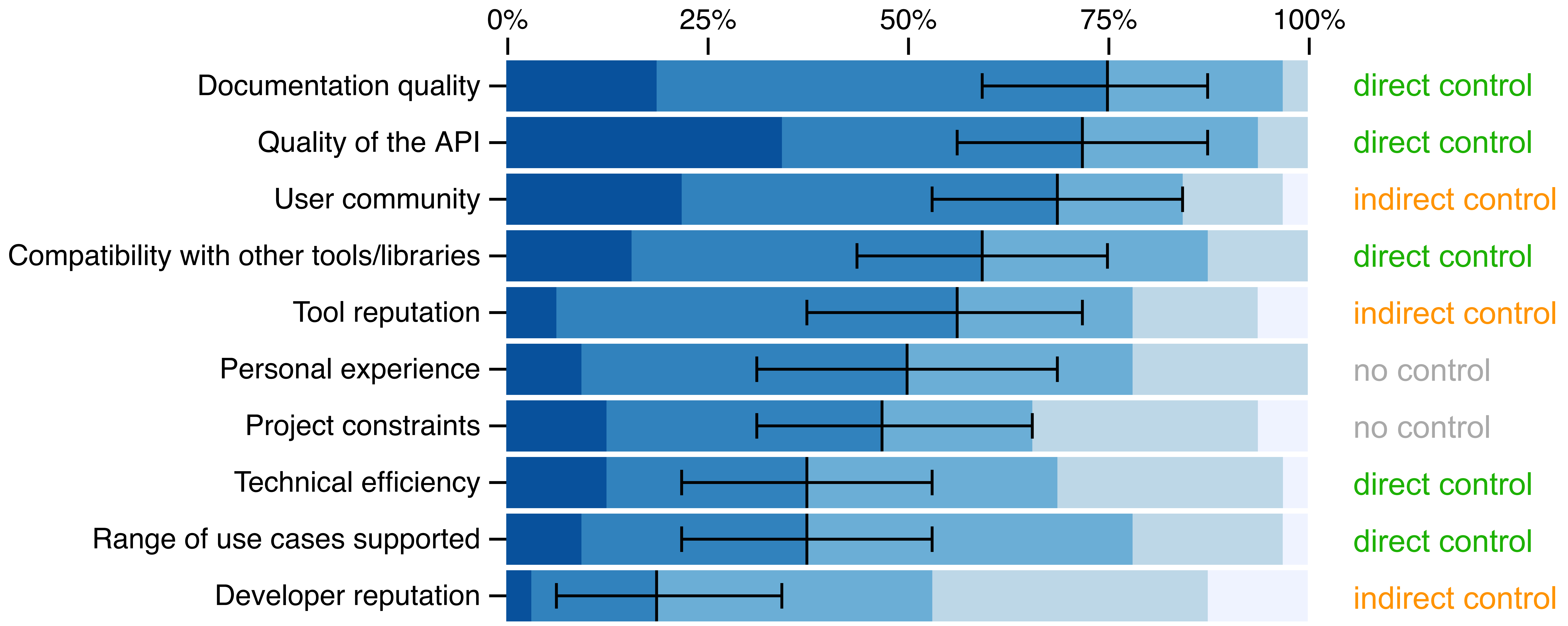
The choice of a library is mostly based on its ease of use, and is directly controlled by its authors



Interviews & Survey

Observation 2

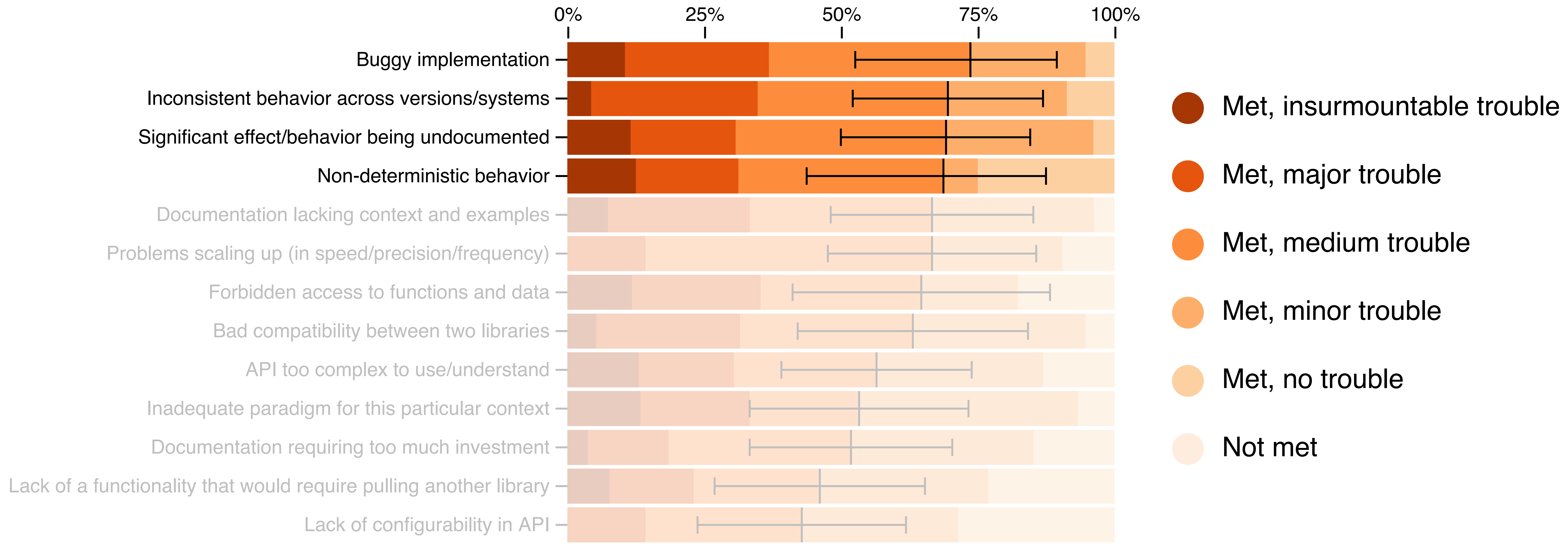
The choice of a library is mostly based on its ease of use, and is directly controlled by its authors



Interviews & Survey

Observation 3

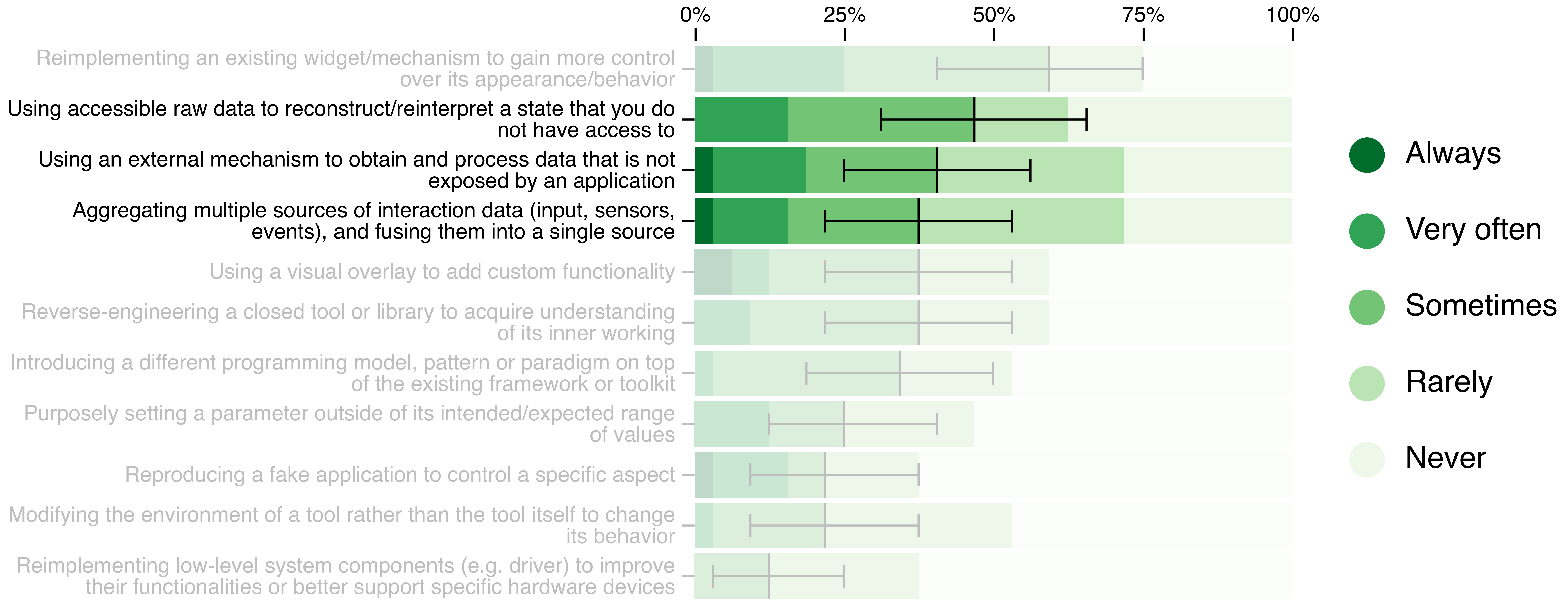
Unpredictability is the most critical problem experienced by researchers with interaction libraries



Interviews & Survey

Observation 4

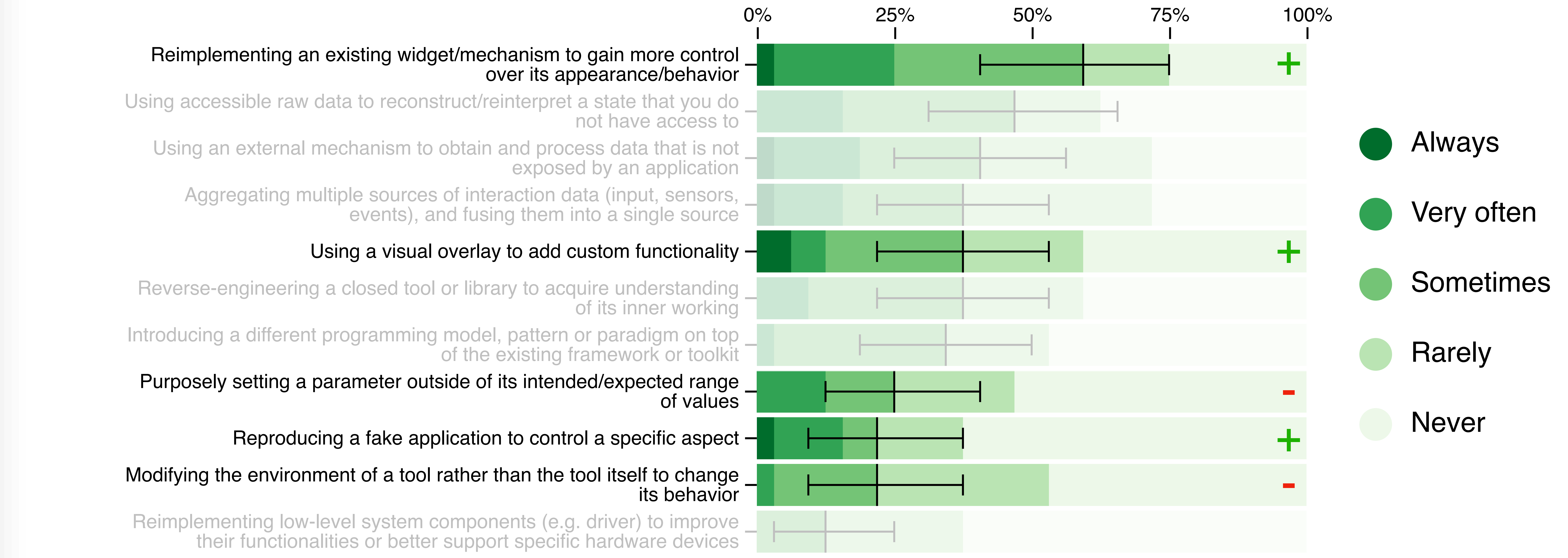
Strategies for gathering and processing interaction data are among the most frequent for our participants



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)

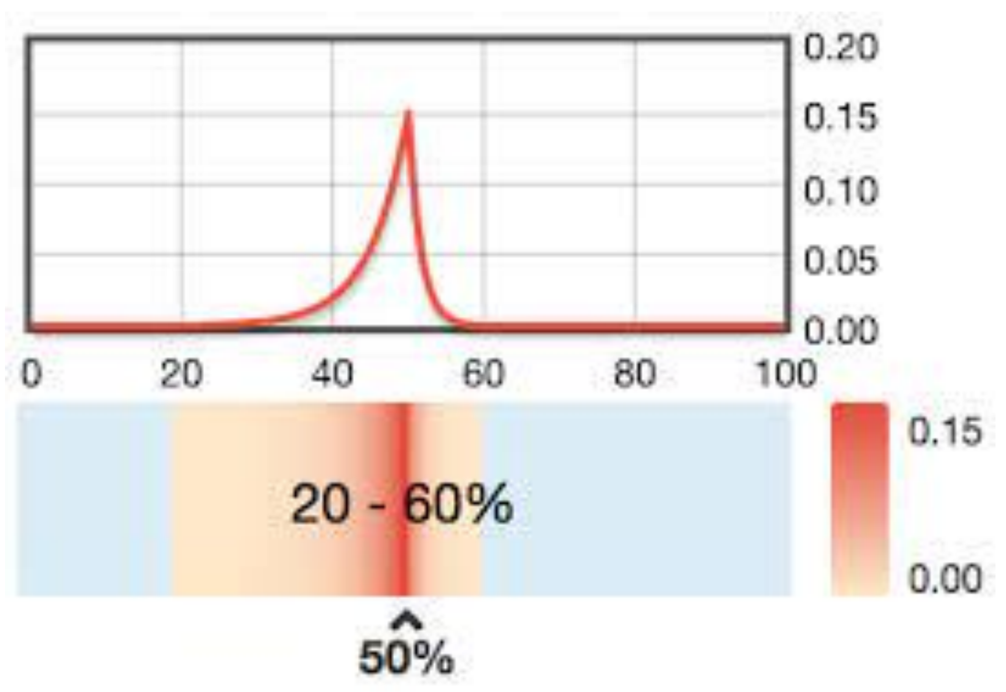
Design recommendations

Duplicate

Allow the duplication of singular elements to foster opportunities for extensions

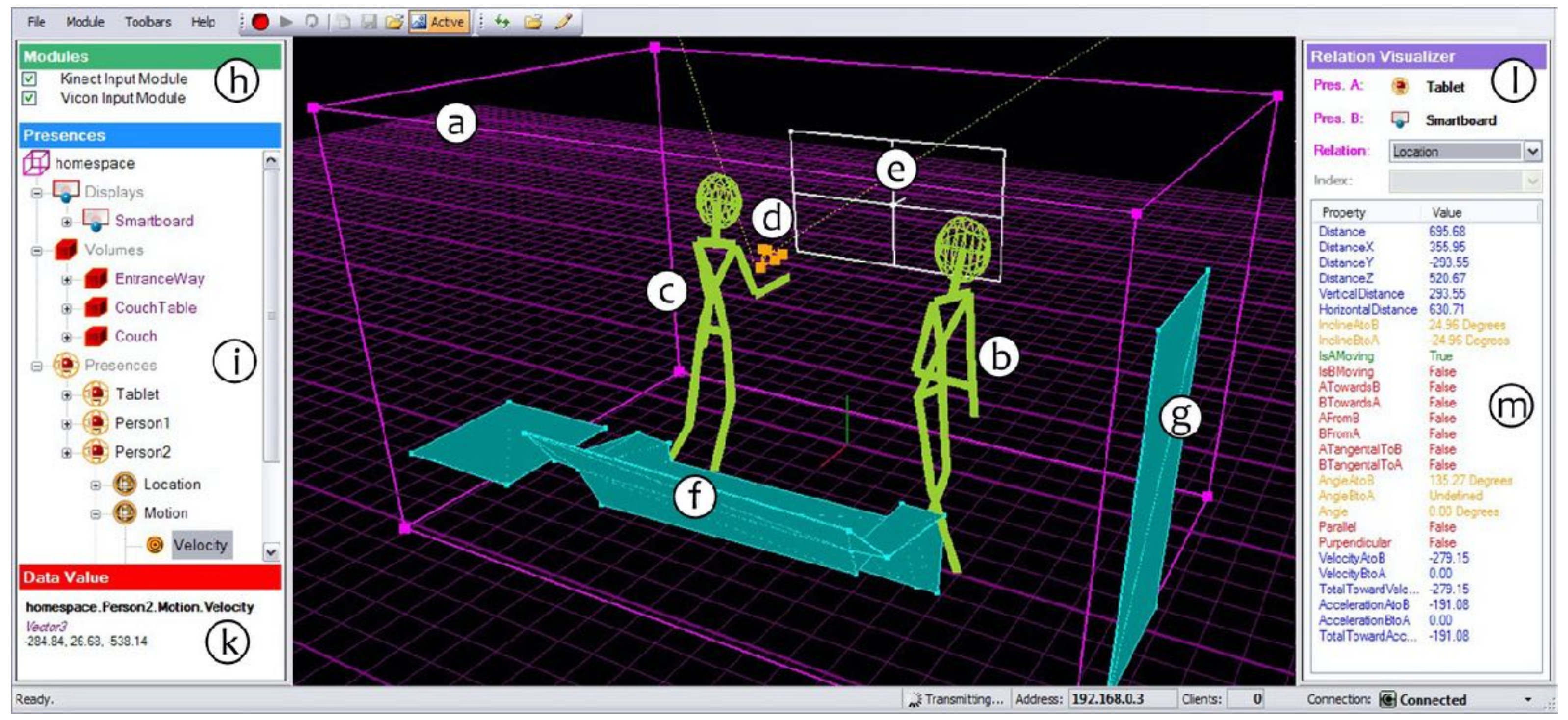
<u>Method</u>: for each element/property/argument

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.)

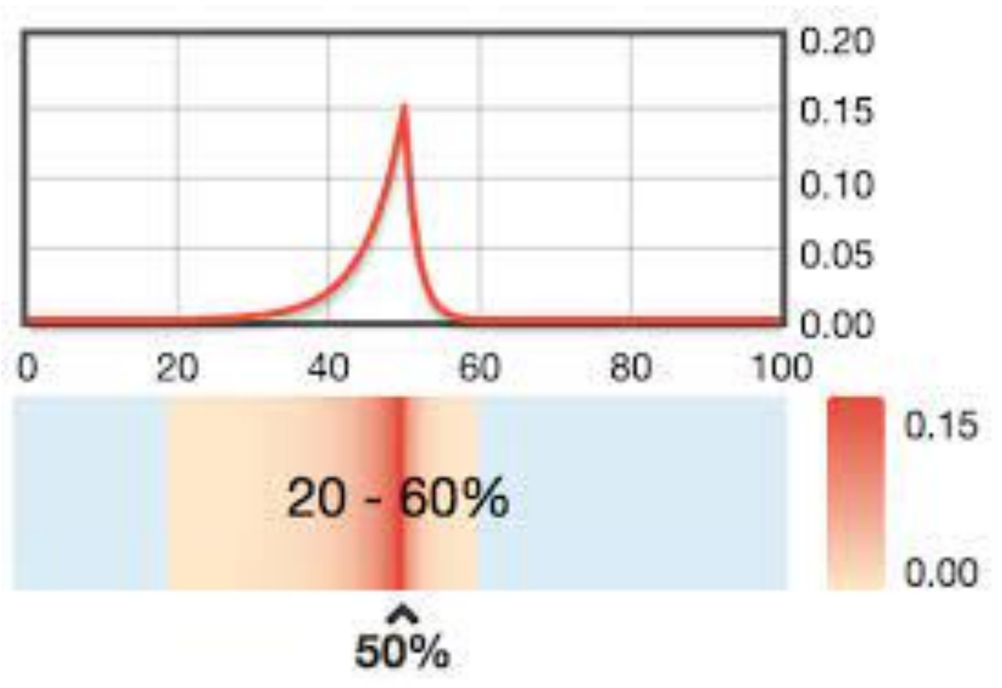

Proximity Toolkit (Marquardt et al.)

Design recommendations

Duplicate

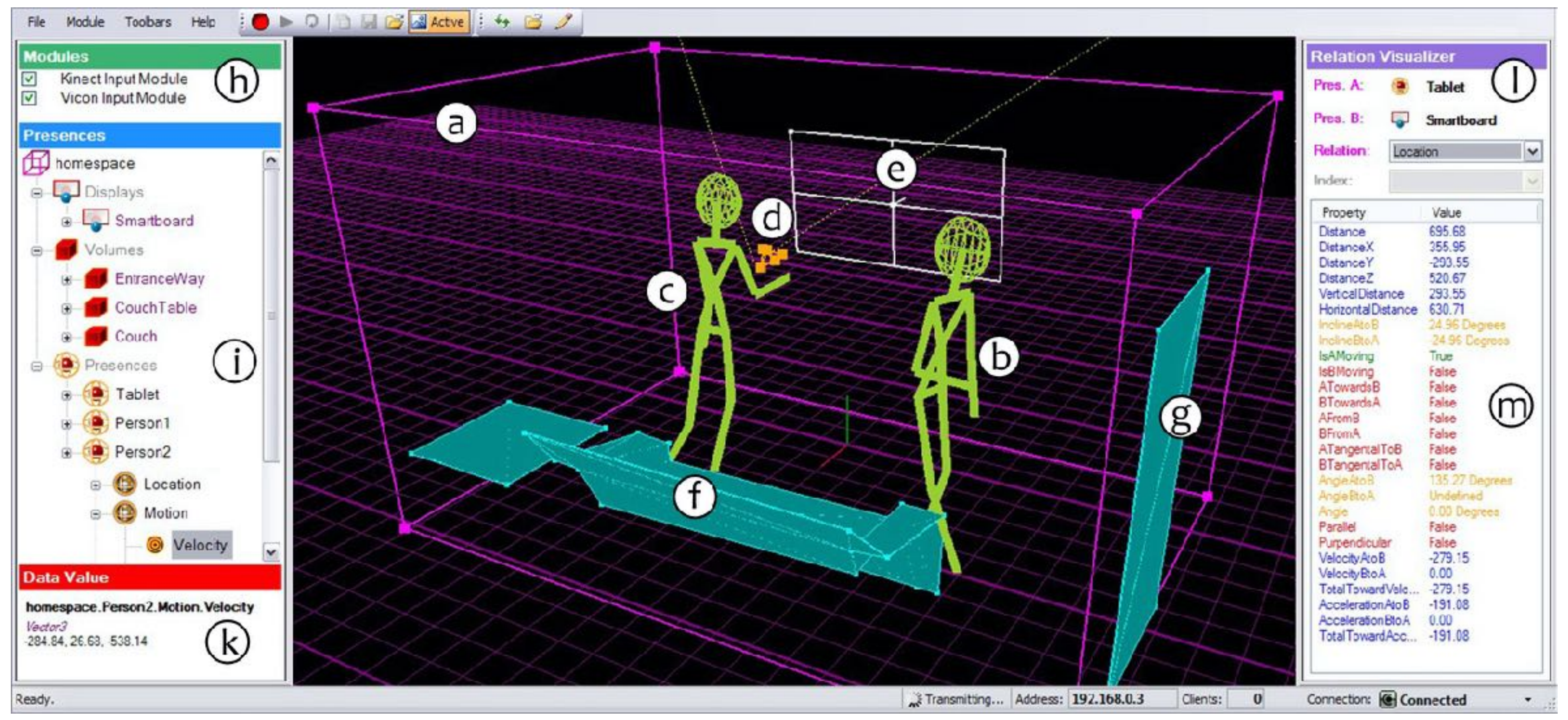
Do not implement these examples → finer reuse/composition

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



Probability Distribution Sliders (Greis et al.)

ExposeHK (Malacria et al.)


Proximity Toolkit (Marquardt et al.)

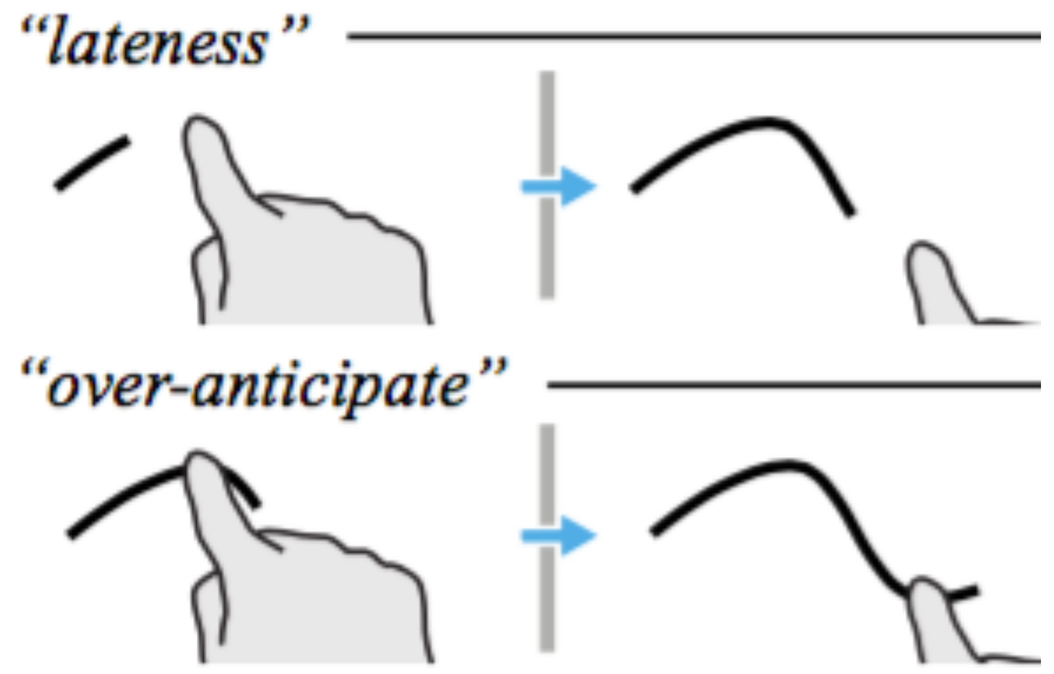
Design recommendations

Accumulate

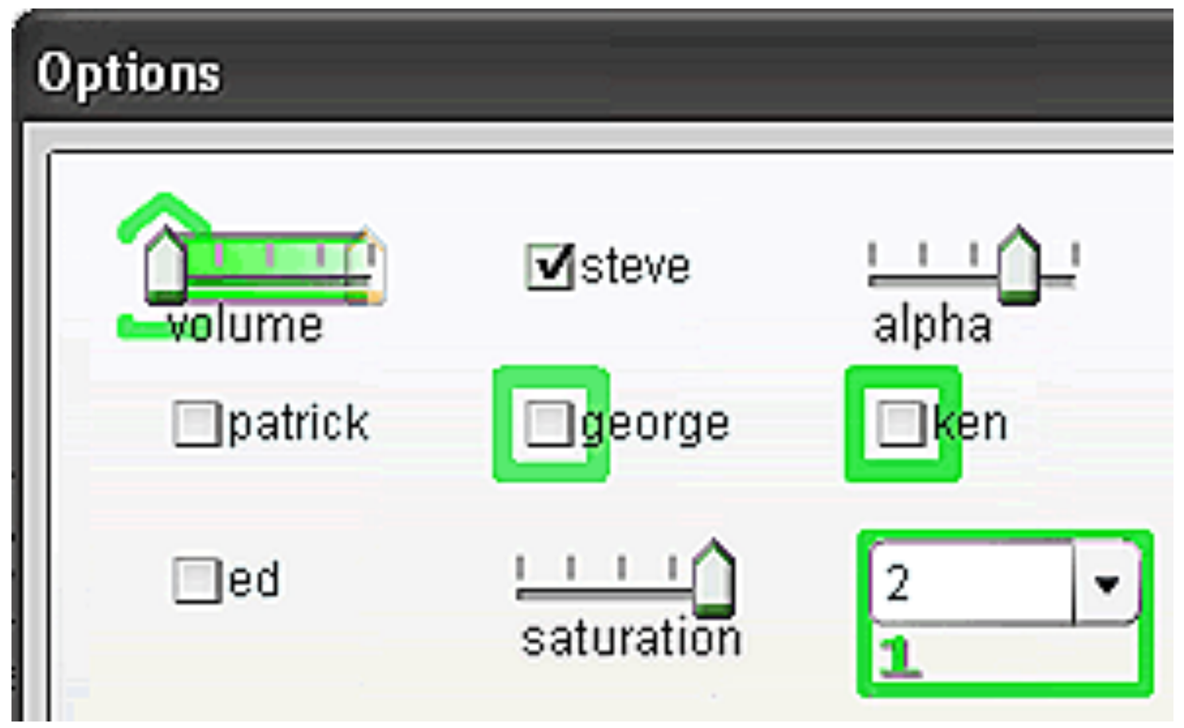
Accumulate rather than replace to keep a history of changes

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

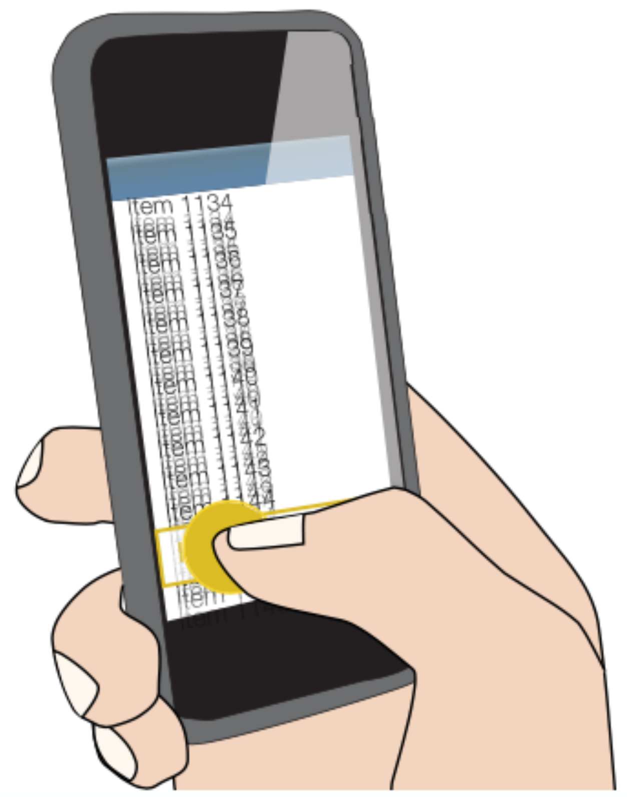
1) Is this data replaced by another?
2) Could it make sense to keep both at any time?



TurboTouch (Nancel et al.)



Phosphor (Baudisch et al.)

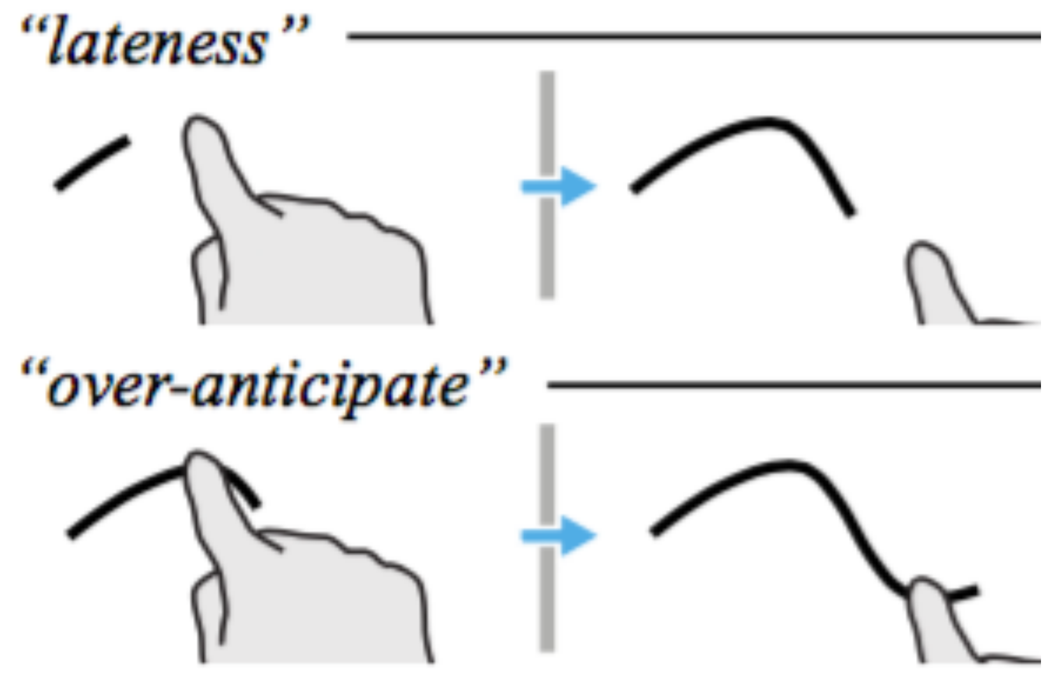

ForceEdge (Antoine et al.)

Design recommendations

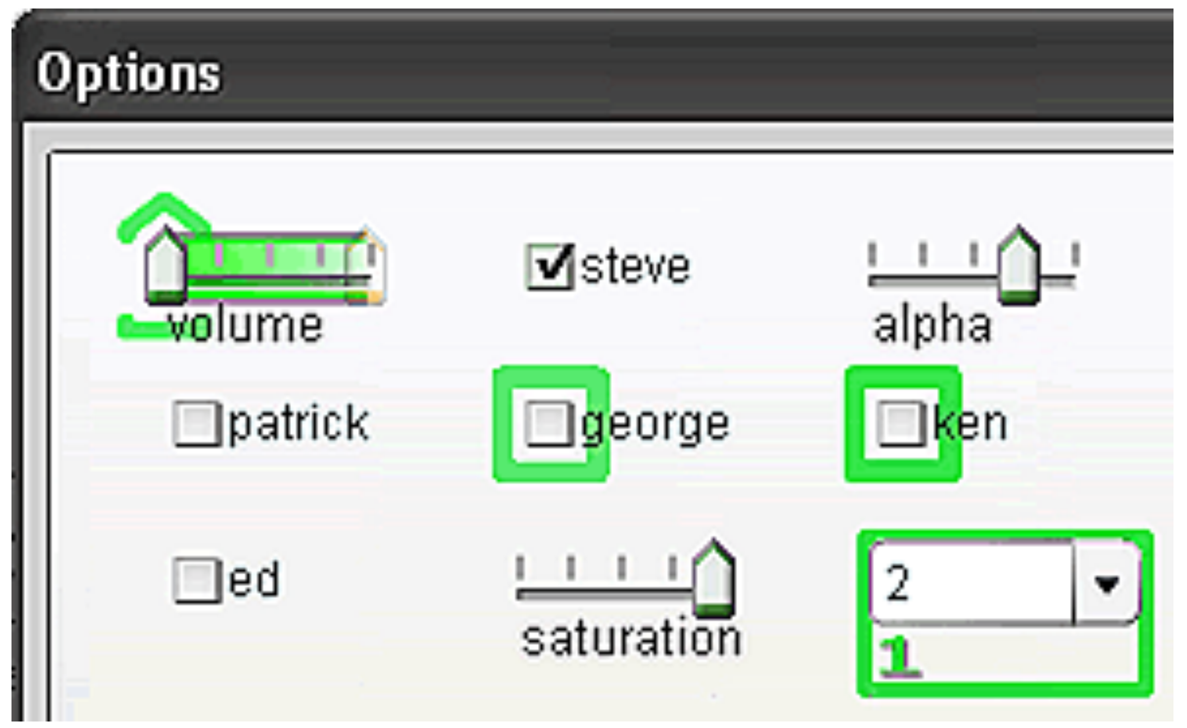
Accumulate

Accumulation over time/space

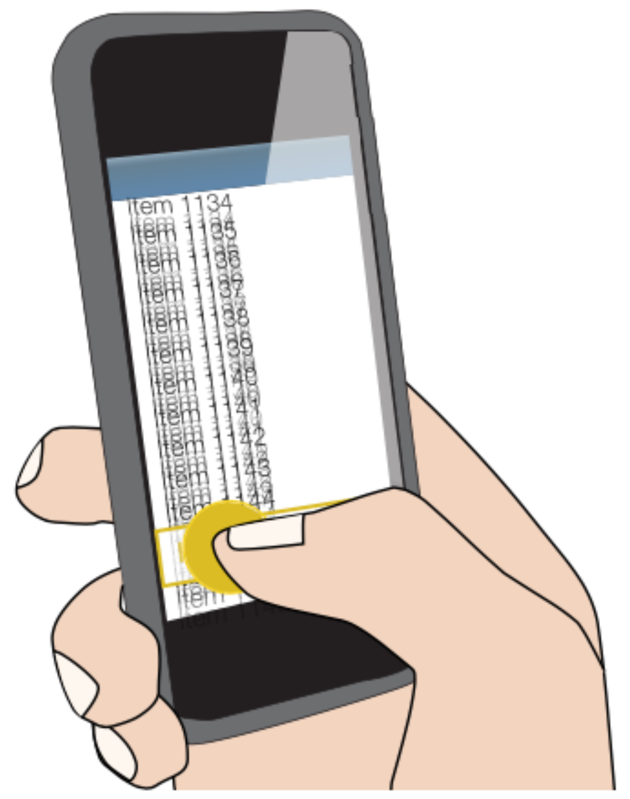
Polymorphism



TurboTouch (Nancel et al.)



Phosphor (Baudisch et al.)



ForceEdge (Antoine et al.)

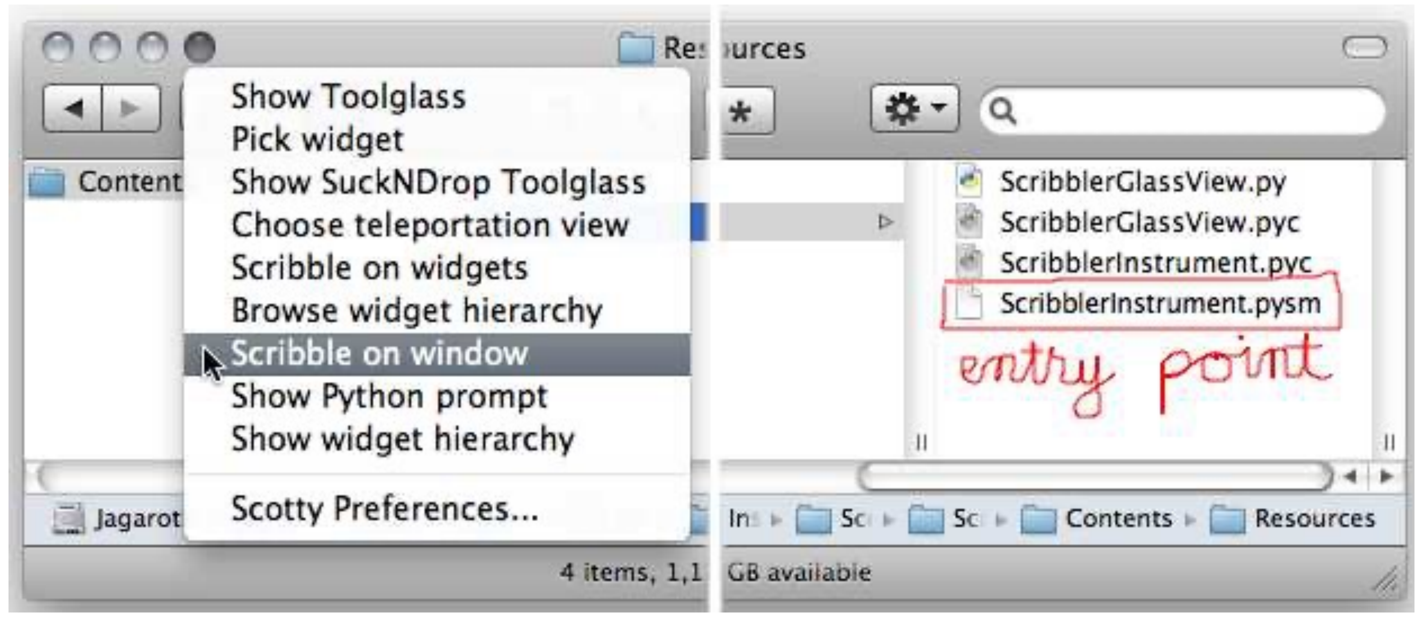
Design recommendations

Defer

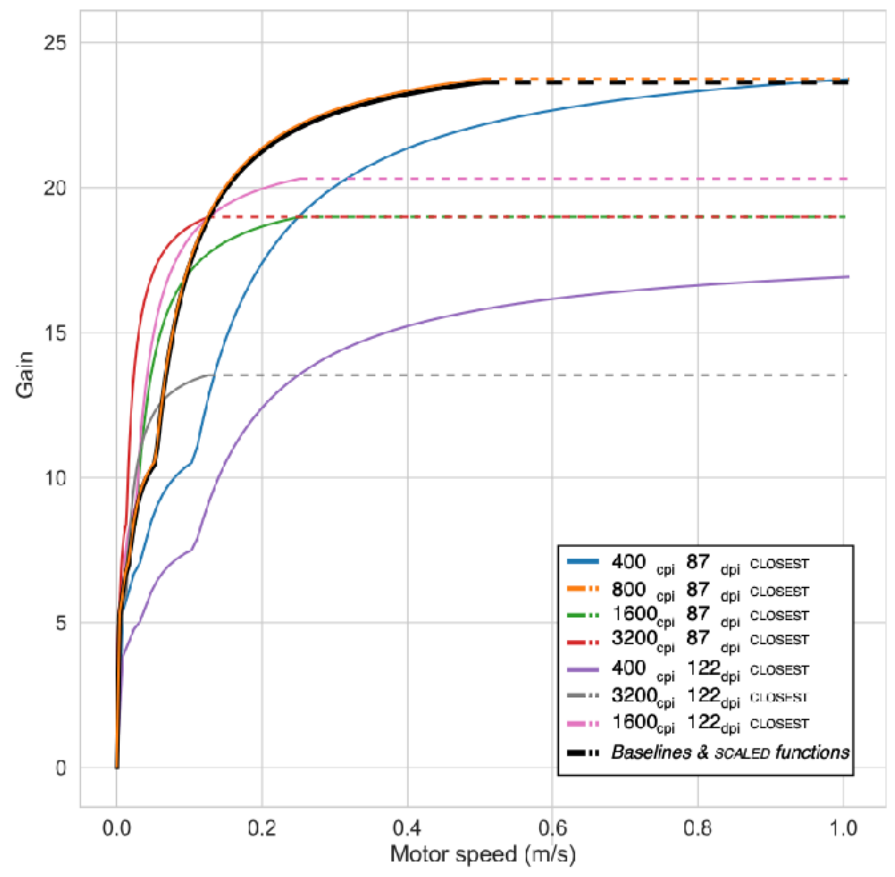
Defer the execution of predefined behaviors to enable their monitoring and replacement

Method: for each function/method

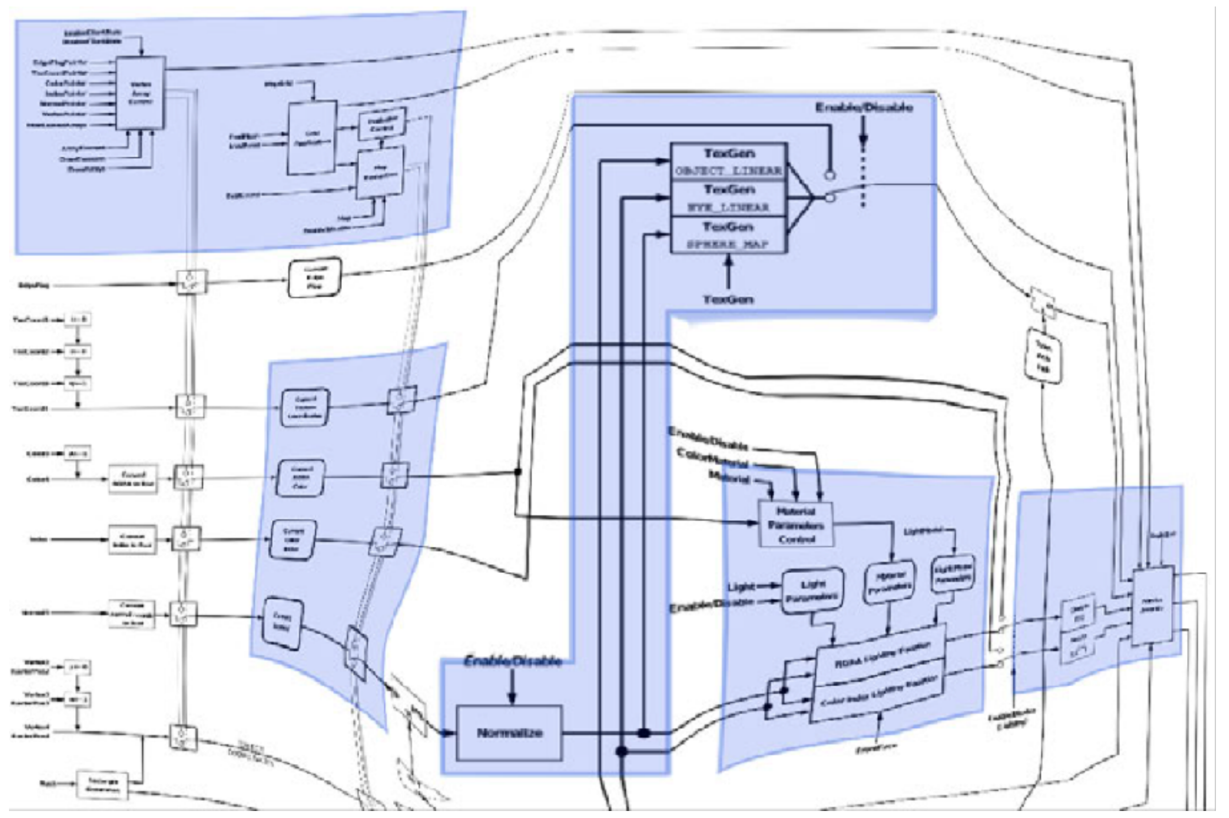
1) Can this action be intercepted? (i.e. canceled, altered or repeated)
2) If not, could it be useful at run-time or compile-time?



Scotty (Eagan et al.)



libpointing (Casiez et al.)



JellyLens (Pindat et al.)

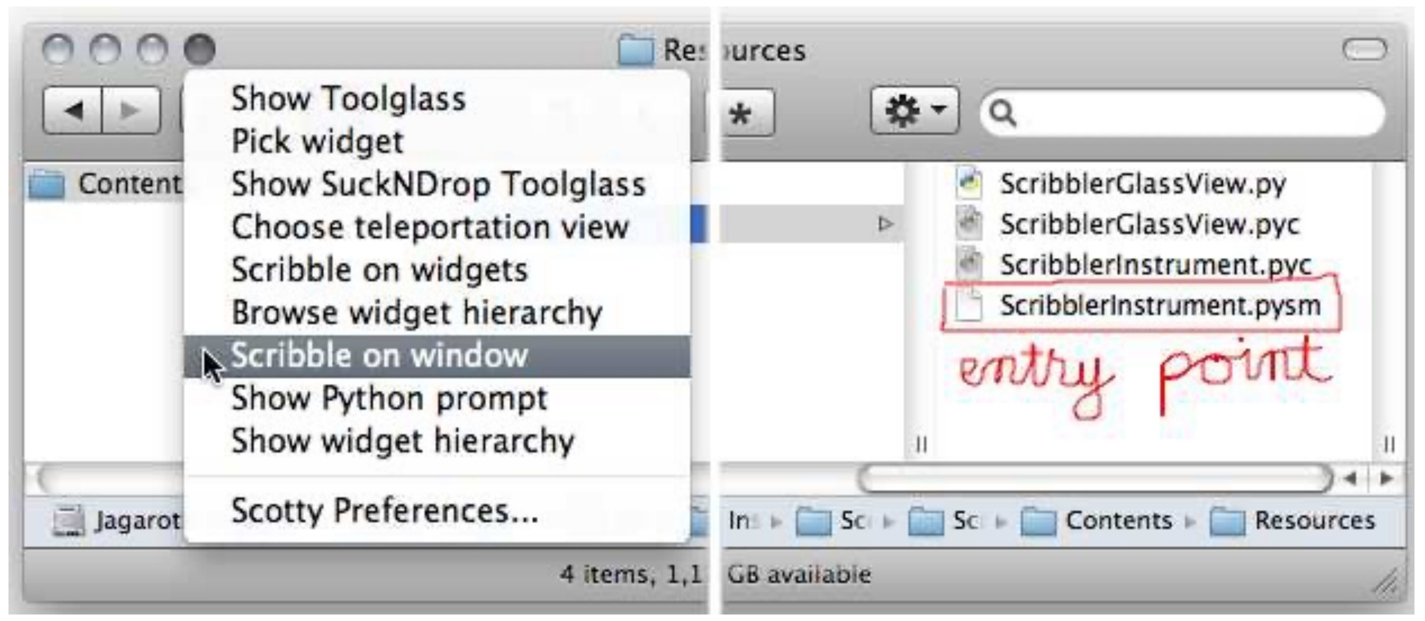
Design recommendations

Defer

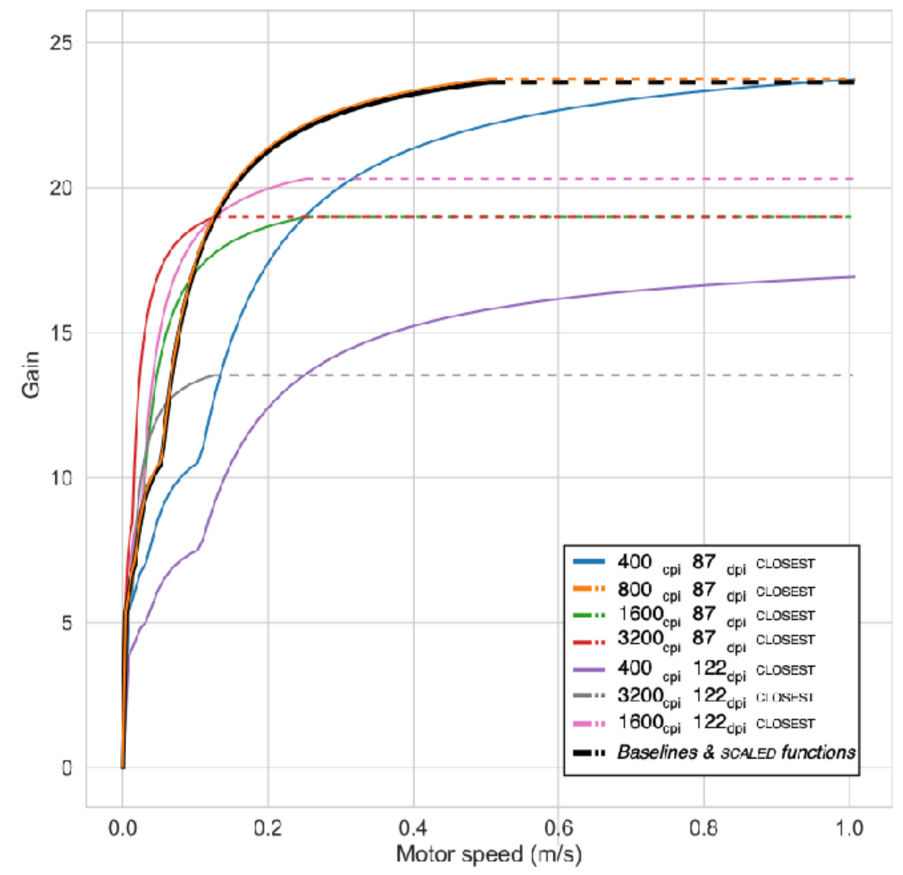
Split commands into (i) placing an order and (ii) executing it

More scalable indirection mechanisms:

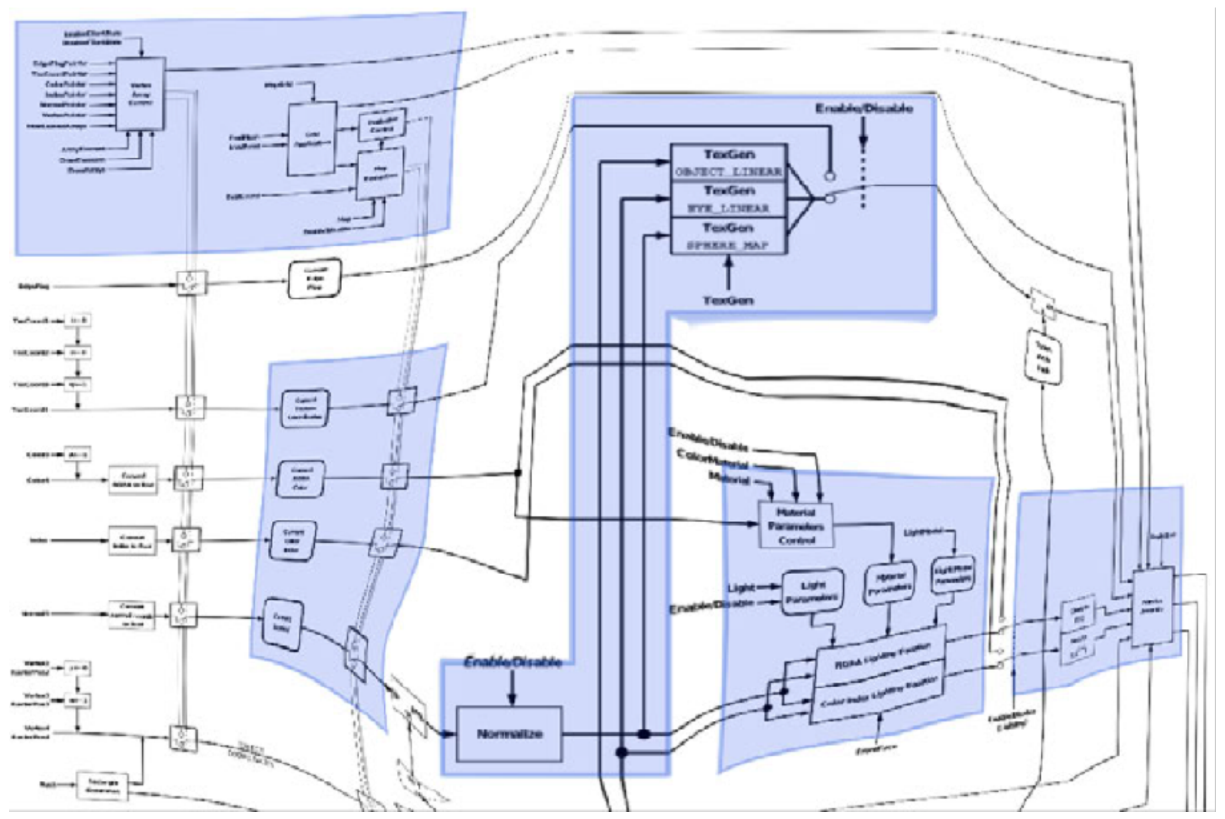
- open intermediate structures (e.g. DOM, framebuffer)
- software buses



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

Interviews & Survey

Interviews

9 HCI researchers (+1 pilot)
6 Seniors researchers
1 Engineer
1 PhD student
1 Master student

Semi-structured
2~4 past projects
Problems faced
Typical development cycle

in-situ interviews
1 interviewer, 1 participant
audio recording

$$\text{audio 9h39} \xrightarrow{\text{transcription}} \text{2519 lines, 9 files} \xrightarrow{\text{extraction}} \text{228 codes} \xrightarrow{\text{analysis}} \text{48 themes}$$

audio
9h39

transcription

2519 lines
9 files

extraction

228 codes
- problems
- strategies
- utilities
- needs

analysis

48 themes
- problems
- strategies
- utilities

Interviews & Survey

Survey

32 participants (+4 pilot)
2/3 code < 40% of their time
2/3 advanced or expert

online questionnaire
rating relevance of items
20 minutes

chi-announcements@acm.org
2nd batch to former teams

- What are the most important criteria for choosing interaction libraries? (R1)
- What are the most limiting implementation problems for researchers? (R2)
- Which strategies are most used to circumvent and overcome these problems? (R3)

Interviews & Survey

Limits & Scope

Interviews with local team of researchers → risk of missing some problems/strategies

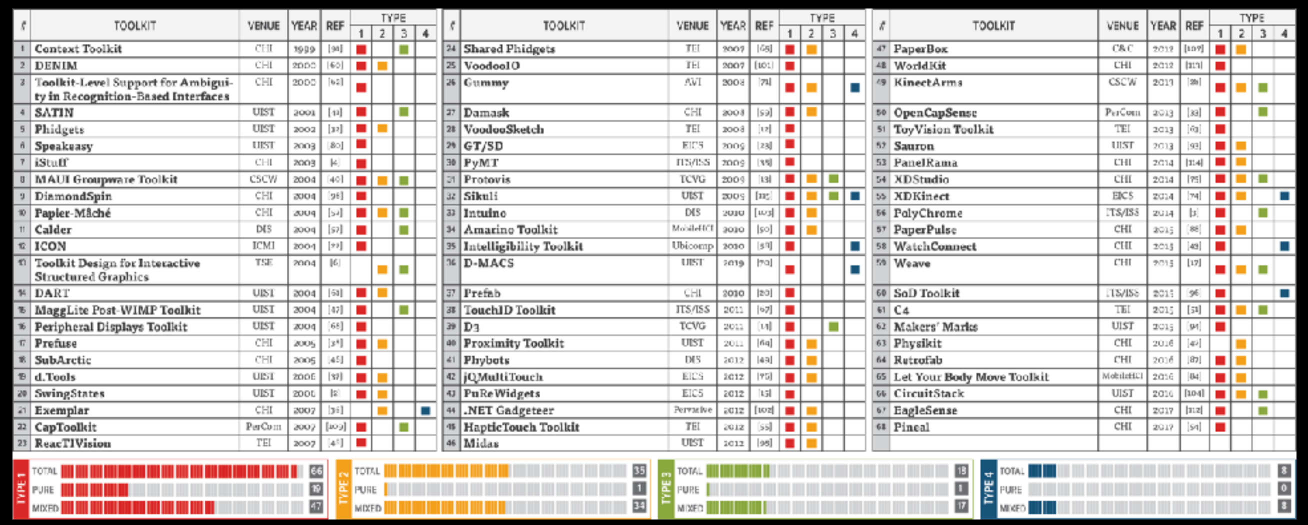
Being familiar with the interviewees → risk of overestimating the severity of problems

Lack of baseline survey with non-researchers → lack of emphasis on the uniqueness of research needs

Scope: understand why researchers are unsatisfied & suggest directions of improvement

Design recommendations

Rationales from toolkits



#	TOOLKIT	VENUE	YEAR	REF	TYPE 1	TYPE 2	TYPE 3	TYPE 4
1	Context Toolkit	CHI	1999	[94]	🟥		🟩	
2	DENIM	CHI	2000	[60]	🟥	🟧		
3	Toolkit-Level Support for Ambiguity in Recognition-Based Interfaces	CHI	2000	[63]	🟥			
4	SATIN	UIST	2001	[41]	🟥		🟩	
5	Phidgets	UIST	2002	[32]	🟥	🟧		
6	Speakeasy	UIST	2003	[80]	🟥			
7	iStuff	CHI	2003	[4]	🟥			
8	MAUI Groupware Toolkit	CSCW	2004	[40]	🟥	🟧	🟩	
9	DiamondSpin	CHI	2004	[98]	🟥			
10	Papier-Mâché	CHI	2004	[54]	🟥	🟧	🟩	
11	Calder	DIS	2004	[57]	🟥		🟩	
12	ICON	ICMI	2004	[22]	🟥			
13	Toolkit Design for Interactive Structured Graphics	TSE	2004	[6]		🟧	🟩	
14	DART	UIST	2004	[53]	🟥	🟧		
15	MaggLite Post-WIMP Toolkit	UIST	2004	[47]	🟥		🟩	
16	Peripheral Displays Toolkit	UIST	2004	[68]	🟥			
17	Prefuse	CHI	2005	[38]	🟥	🟧		
18	SubArctic	CHI	2005	[45]	🟥			
19	d.Tools	UIST	2006	[37]	🟥	🟧		
20	SwingStates	UIST	2006	[2]	🟥	🟧		
21	Exemplar	CHI	2007	[36]		🟧		🟦
22	CapToolkit	PerCom	2007	[105]	🟥		🟩	
23	ReacTIVision	TEI	2007	[43]	🟥			
24	Shared Phidgets	TEI	2007	[65]	🟥	🟧		
25	VoodooIO	TEI	2007	[101]	🟥			
26	Gummy	AVI	2008	[71]	🟥	🟧		🟦
27	Damask	CHI	2008	[59]	🟥	🟧		
28	VoodooSketch	TEI	2008	[12]	🟥			
29	GT/SD	EICS	2009	[23]	🟥			
30	PyMT	ITS/ISS	2009	[35]	🟥			
31	Protovis	TCVG	2009	[13]	🟥	🟧	🟩	
32	Sikuli	UIST	2009	[115]	🟥	🟧	🟩	🟦
33	Intuino	DIS	2010	[103]	🟥	🟧		
34	Amarino Toolkit	MobileHCI	2010	[50]	🟥	🟧		
35	Intelligibility Toolkit	Ubicomp	2010	[58]	🟥			🟦
36	D-MACS	UIST	2010	[70]	🟥			🟦
37	Prefab	CHI	2010	[20]	🟥			
38	TouchID Toolkit	ITS/ISS	2011	[67]	🟥			
39	D3	TCVG	2011	[14]	🟥		🟩	
40	Proximity Toolkit	UIST	2011	[64]	🟥	🟧		
41	Phybots	DIS	2012	[49]	🟥	🟧		
42	jQMultiTouch	EICS	2012	[75]	🟥	🟧		
43	PuReWidgets	EICS	2012	[15]	🟥			
44	.NET Gadgeteer	Pervasive	2012	[102]	🟥	🟧		
45	HapticTouch Toolkit	TEI	2012	[55]	🟥	🟧		
46	Midas	UIST	2012	[98]	🟥	🟧		
47	PaperBox	C&C	2012	[109]	🟥	🟧		
48	WorldKit	CHI	2012	[111]	🟥			
49	KinectArms	CSCW	2013	[29]	🟥	🟧	🟩	
50	OpenCapSense	PerCom	2013	[33]	🟥		🟩	
51	ToyVision Toolkit	TEI	2013	[63]	🟥			
52	Sauron	UIST	2013	[93]	🟥	🟧		
53	PanelRama	CHI	2014	[114]	🟥	🟧		
54	XDStudio	CHI	2014	[75]	🟥	🟧	🟩	
55	XDKinect	EICS	2014	[74]	🟥	🟧		🟦
56	PolyChrome	ITS/ISS	2014	[3]	🟥		🟩	
57	PaperPulse	CHI	2015	[88]	🟥	🟧		
58	WatchConnect	CHI	2015	[49]	🟥			🟦
59	Weave	CHI	2015	[17]	🟥	🟧	🟩	
60	SoD Toolkit	ITS/ISS	2015	[96]	🟥			🟦
61	C4	TEI	2015	[51]	🟥	🟧	🟩	
62	Makers' Marks	UIST	2015	[94]	🟥			
63	Physikit	CHI	2016	[42]		🟧		
64	Retrofab	CHI	2016	[87]	🟥	🟧		
65	Let Your Body Move Toolkit	MobileHCI	2016	[84]	🟥	🟧		
66	CircuitStack	UIST	2016	[104]	🟥	🟧	🟩	
67	EagleSense	CHI	2017	[112]	🟥		🟩	
68	Pineal	CHI	2017	[54]	🟥			

	TYPE 1	TYPE 2	TYPE 3	TYPE 4
TOTAL	66	35	18	8
PURE	19	1	1	0
MIXED	47	34	17	8

Evaluation Strategies for HCI Toolkit Research (Ledo et al.)

Example in D3 (Bostock et al., 2011):

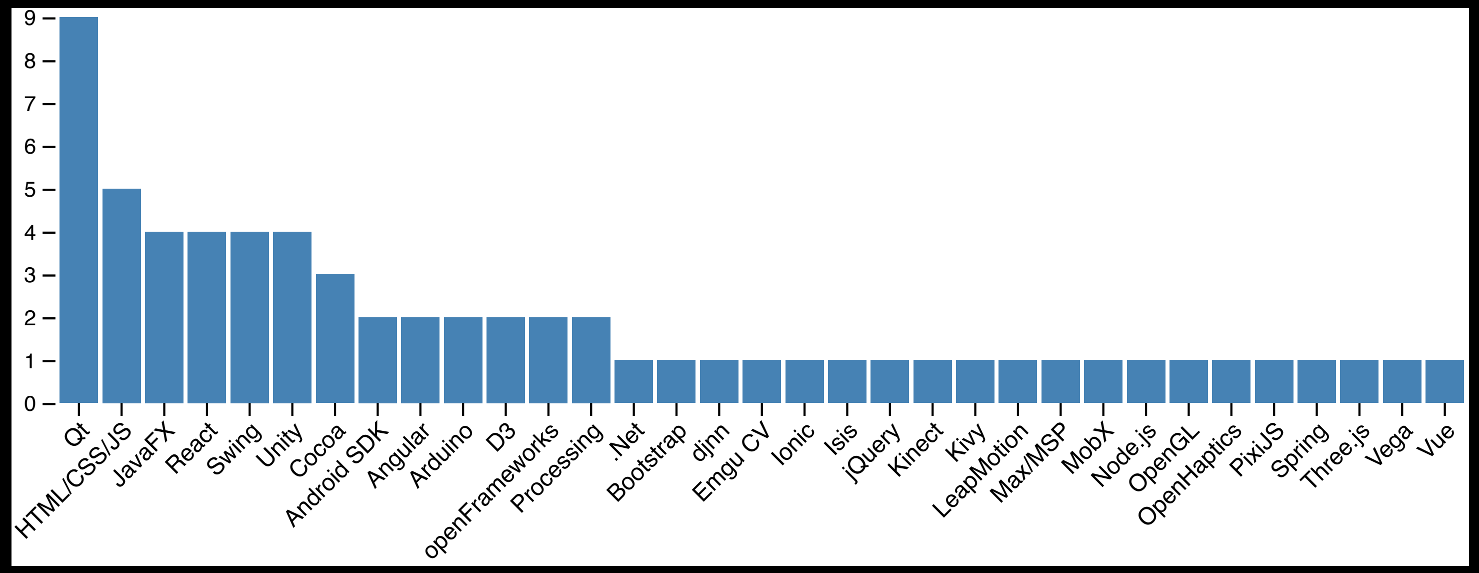
- when a scene is generated from data, specify explicit transformations rather than letting the scene be generated implicitly
- the update of a property depending on another is immediate rather than deferred to facilitate live inspection and debugging
- intermediate representations rely on existing native formats to leverage existing user knowledge and helper tools

Extracting recommendations for other frameworks/toolkits:

- rarely discussed in papers
- highly contextual
- lack of justification on their positive impact for users

Design recommendations

Rationales from frameworks



Example from Qt (Knoll, 2017):

- APIs that lead to readable and maintainable code
- easy to learn and use but hard to misuse
- performant
- flexible
- keeping it simple
- API stability
- world class tools

Extracting recommendations for other frameworks/toolkits:

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

Design recommendations

Studies on researchers' needs

Example in *Usability requirements for interaction-oriented development tools* (Letondal, 2010):

- minimising information complexity
- minimising access complexity
- minimising unpredictability
- graphics
- runtime adaptation
- interaction modalities
- distribution
- supporting code production
- matching code and execution
- managing the life cycle
- managing reuse and knowledge capitalization
- managing collective development

Extracting recommendations for other frameworks/toolkits:

- good for understanding complexity of frameworks and comparing them
- need more traction to generate more in-depth descriptions