Designing and Programming with Multiplicity

Philip Tchernavskij
March 2020

Overview

The tools I have used to develop design problems and propose interventions in the context of multiple computing

- I'm still developing and making sense of some

- Maybe some are useful to IDRC/the Fluid community

- I would love to hear about your theories and methods as well

- Please ask questions!

From human factors to human actors

Early Human-Computer Interaction (HCI) models described humans as components of computer systems, leading to

- Lab studies defining parameters of "human factors"

- Concept of users without history, context, or expertise

From human factors to human actors
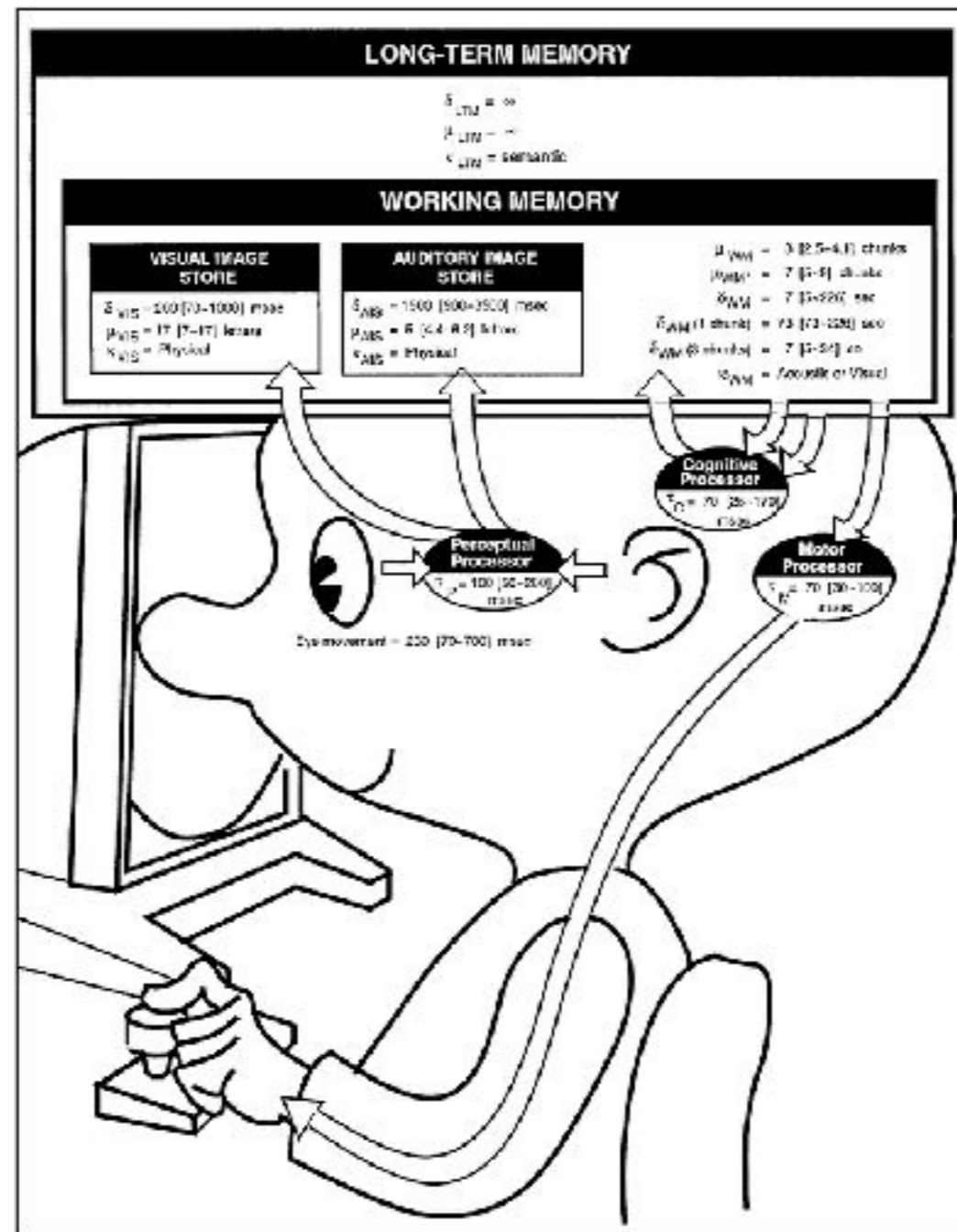
From human factors to human actors

Second-wave HCI drew from methods and theories that aimed to describe practices and context more holistically, e.g. ethnography
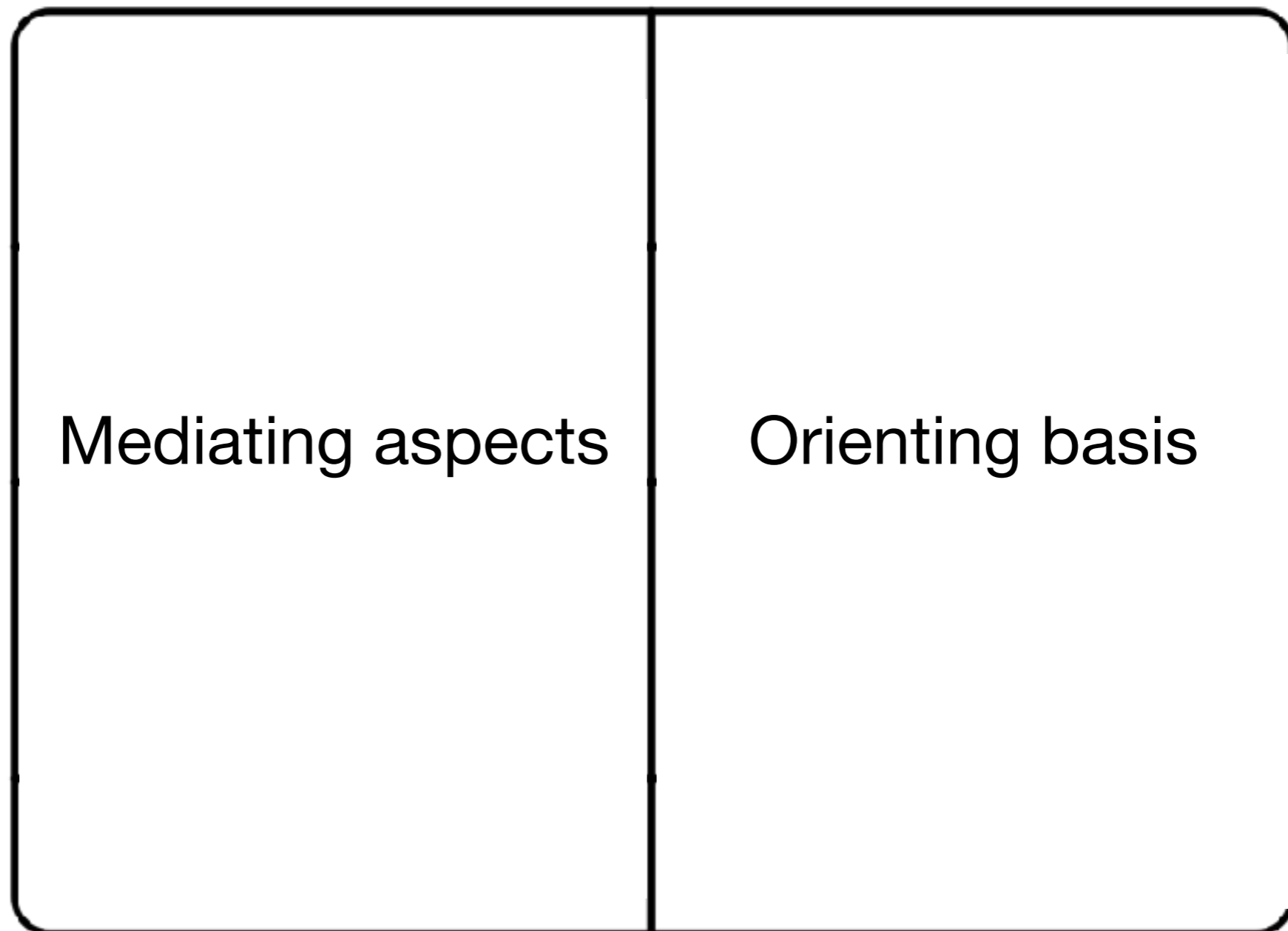
The Human-Artifact Model

Tool for analysis and design developed by Bødker & Klokmose

- Based on cultural-historical activity theory

- Describes human-artifact (mis-)match

- Views interaction as dialectical and developing, not causal and static

- Combines expertise, ability, and situational context

The Human-Artifact Model

Artifact | Human

Mediating aspects	Orienting basis

The Human-Artifact Model

	Artifact	Human
What?	Instrumental aspects	Goal-orientation

The Human-Artifact Model

	Artifact	Human
What?	Instrumental aspects	Goal-orientation
How?	Operational aspects - Handling aspects - Adaptive aspects	Operational orientation - Learned handling - Adaptation

The Human-Artifact Model

	Artifact	Human
Why?	Motivational aspects	Motivational orientation
What?	Instrumental aspects	Goal-orientation
How?	Operational aspects - Handling aspects - - - - - - - - - - - - Adaptive aspects	Operational orientation - Learned handling - - - - - - - - - - - - Adaptation

HAM and Accessible Interfaces

Accessibility adaptation often focuses on the operational level of artifacts

- How does accessibility and inclusiveness address instrumental and motivational aspects?

- How do accessible interfaces support routinizing and abbreviating common interactions?

Analyzing Development
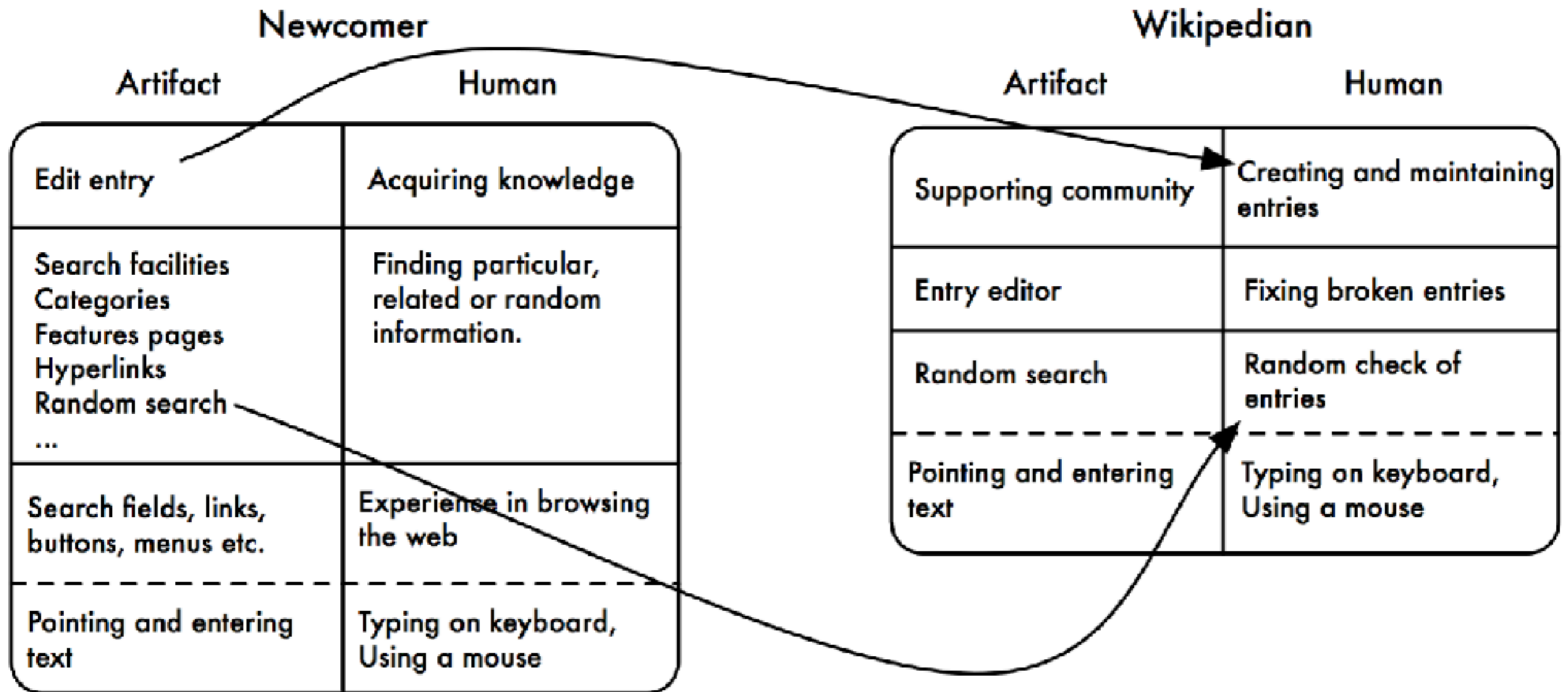
Analyzing Multistability
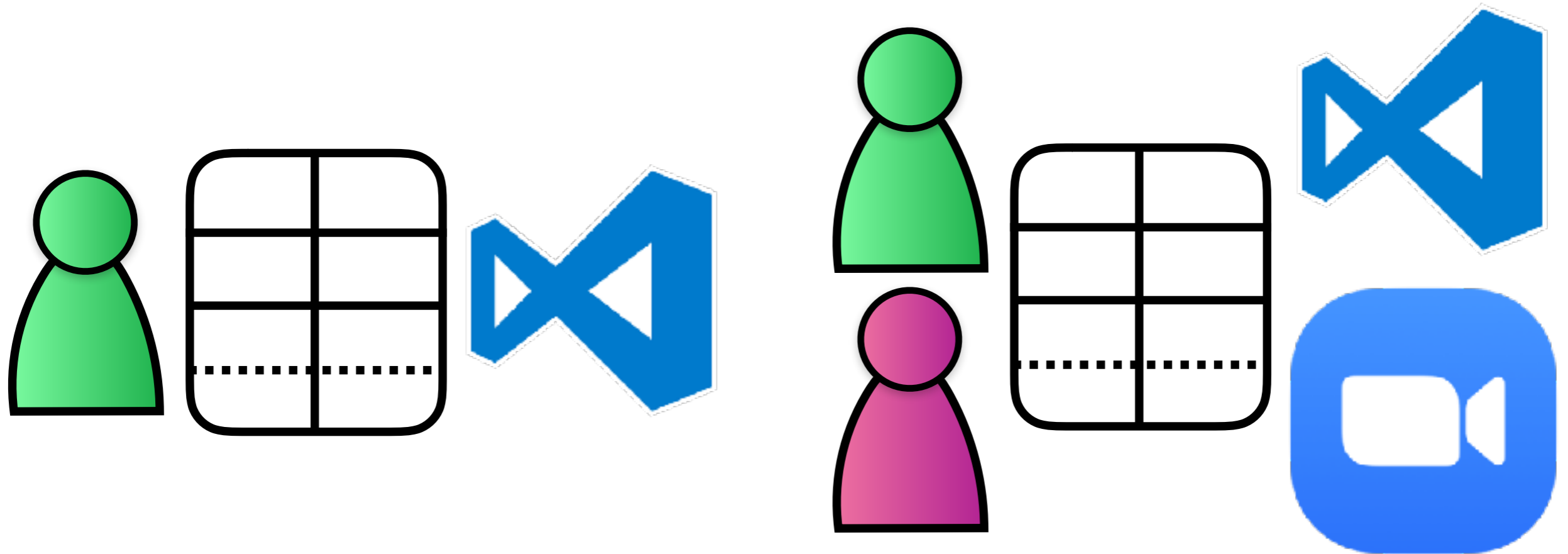
Artifact Ecologies

VSCode, Wikipedia, etc. are not used alone, but alongside other tools, digital and not

- An *artifact ecology* consists of the artifacts that an individual (or community) owns, uses, and has access to

- Overlapping, historical, current, and potential ecologies

  - e.g. the mass of collaboration tools we enumerated last week

How Ecologies Develop

- Happenstance: unpredictable resources, changes, and breakdowns

- Strategies: possibly formalized activities to develop and maintain support for individual or group practice

- Tactics: situated reactions to replace, hack, or reconfigure artifacts in response to happenstance

Appropriation and Design

Happenstance, strategies, and tactics are examples of *intrinsic design*, as opposed to *extrinsic design*

Appropriation and Design

Intrinsic and extrinsic design are entangled

- Adopting commercial and public tools and resources is socially and economically necessary

- Tools "from outside" only become artifacts through *articulation work* that weaves them into a concrete practice

The Limits of Appropriation

- We have theories that describe how people deal with the mess of variously compatible technologies

- But we're missing theories that describe why certain kinds of appropriation happen and others don't

- The tendency towards centralization and enclosure

Qualitative Methods

These theoretical models motivate multiple data gathering methods

- What: critical incident interviews, contextual inquiry

- How: in-situ observations, video analysis

- Why: in-depth interviews, diary studies

- Dynamics and ecologies: **artifact maps** and histories

Artifact Ecology Mapping

Bødker et al. create maps of artifact ecologies to focus on different ecology-level patterns

Artifact Ecology Mapping

Drawing with participants, using prepared stickers

Artifact Ecology Mapping
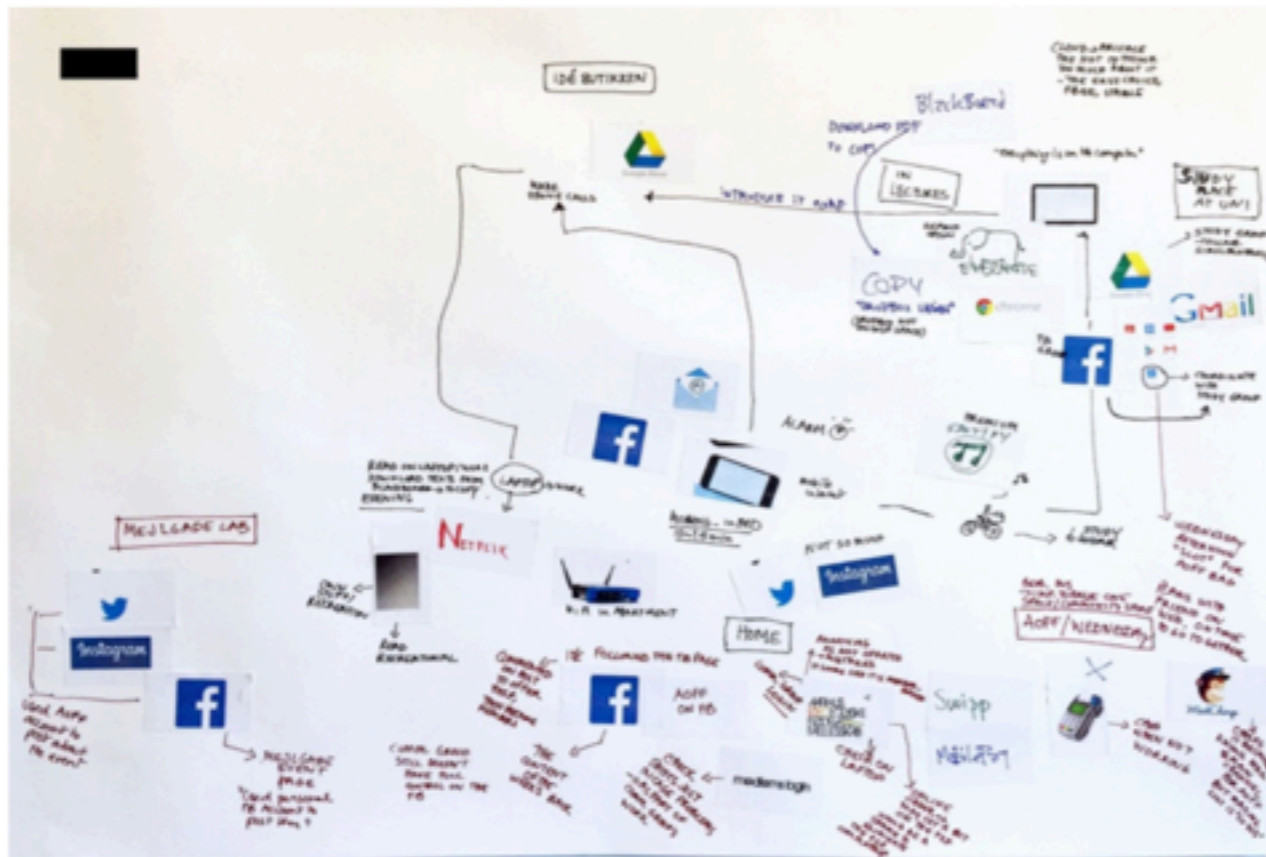
Remapping with co-investigators and annotating

From systems to ecosystems

Our models for the architecture and interfaces of interactive software were also developed in an era of relatively few, relatively stable devices and applications.

From systems to ecosystems

From systems to ecosystems

In the last three decades, the use and production of software has changed substantially

- From the workplace to work and leisure intermixed

- From a single, static system to many heterogeneous devices, apps, and documents

Production tools have been transformed in response, but the user-facing model of applications lags behind

Programming for multiplicity

Tools for customizing and integrating interfaces help people manage a changing and distributed computing environment

- Appropriation work + computation = abbreviation

- Customization supports both diversification and standardization

A customization network
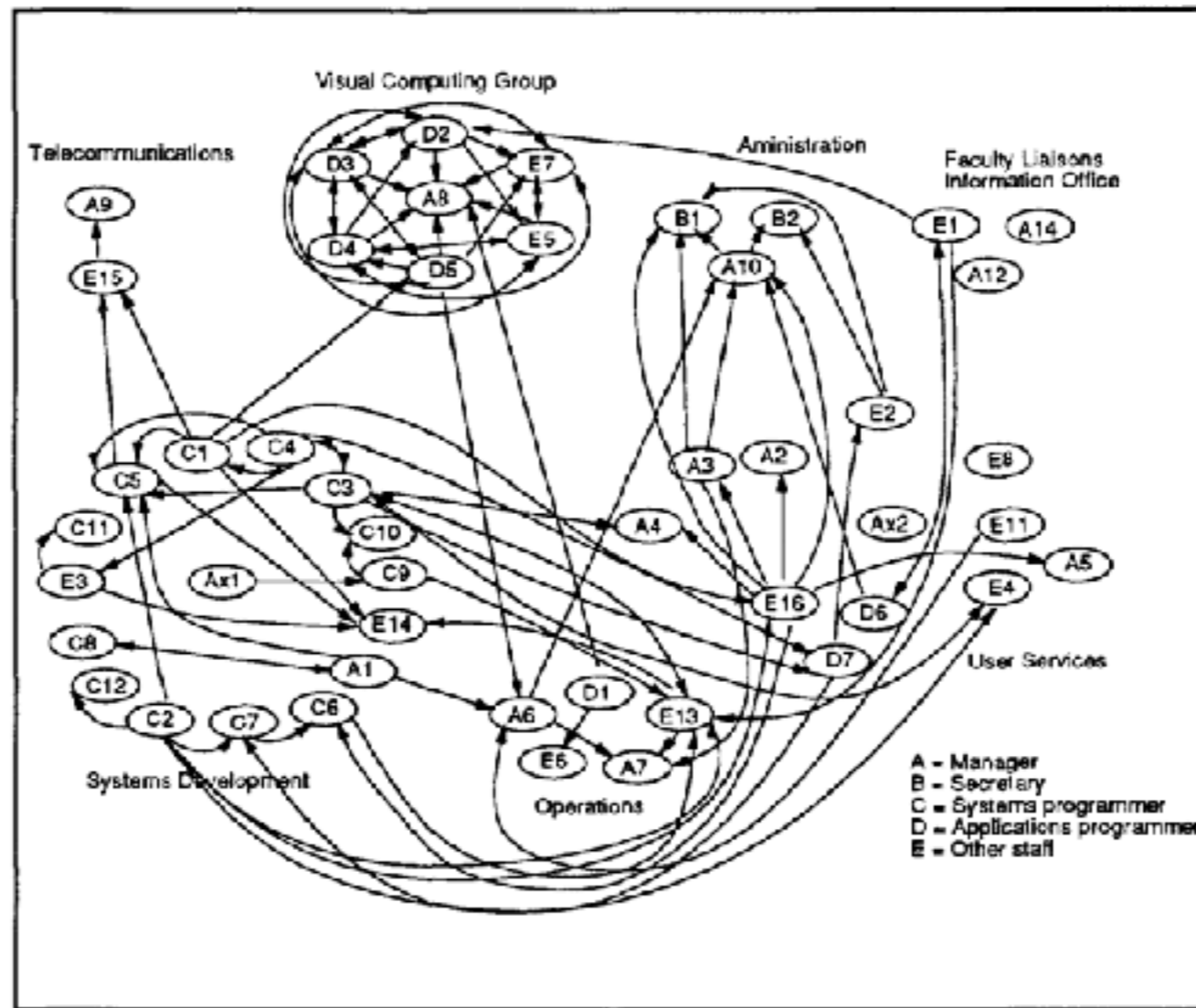
Figure 1: Exchange of Customization Files at Project Athena

(Mackay, 1990)

The Tailorability Mountain and its Inhabitants
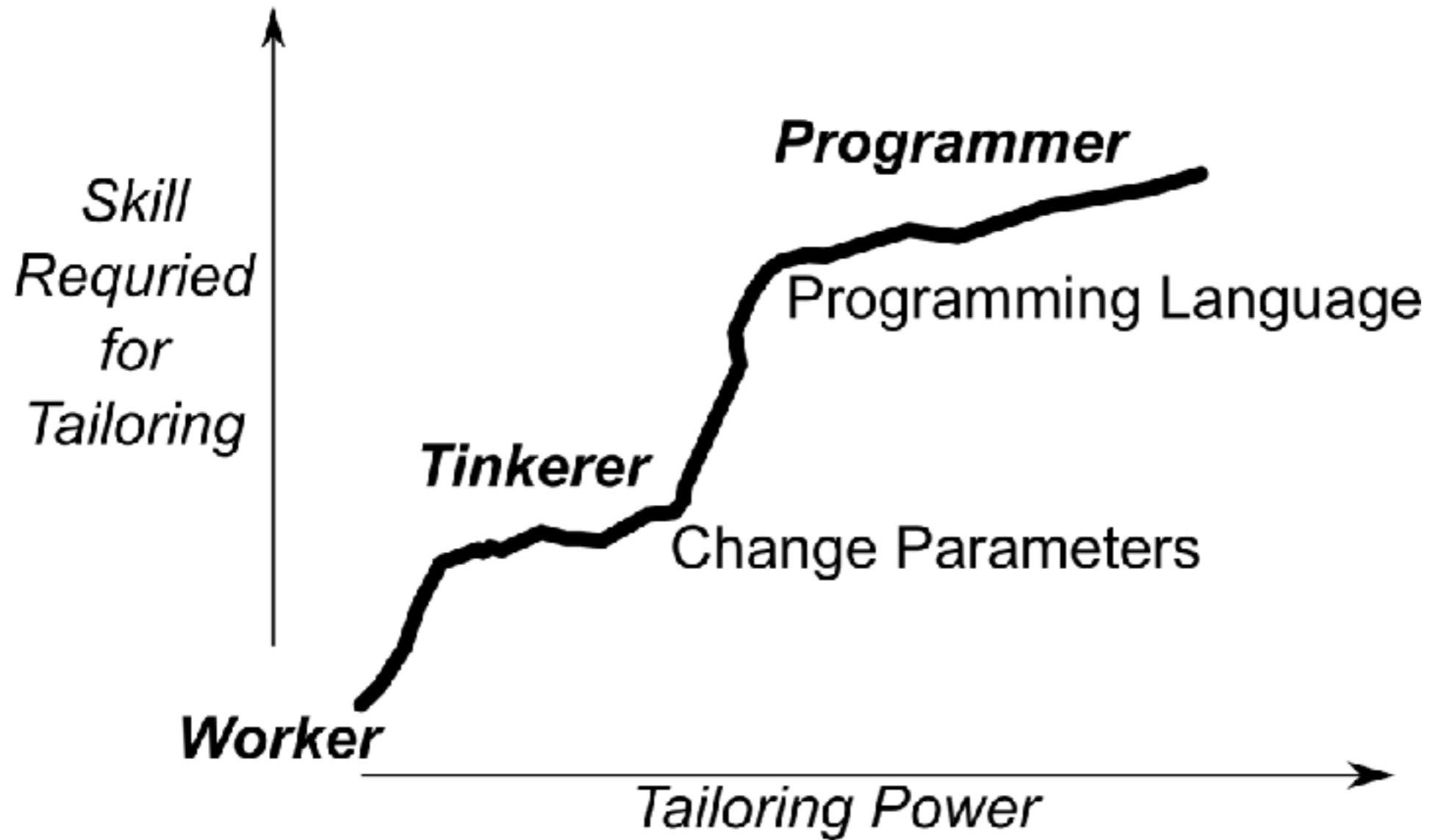
(MacLean et al., 1990)

Buttons - The Gentle Slope to Tailorability and the Folk Who Live on the Hill
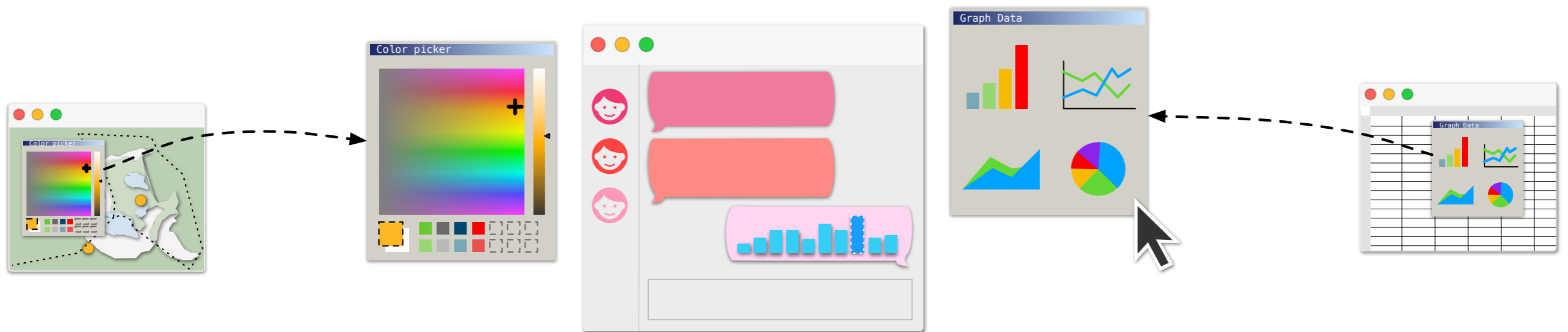

(MacLean et al., 1990)

Original

Malleable Software

Tools such as *If This Then That* enable some ability to integrate applications, but they can't email a button

To enable inclusive networks of customization today, we have to erase the boundaries of applications

Malleable Software

Malleable Software

Malleability could support inclusive design

- Interoperability and customizability support maintenance

- Assembling systems piecemeal means more people can have a say

Conclusion

How do we improve interfaces when problems can't be localized to a single user in front a of a single system at a single point in time?

We need to be able to describe and design for technological practices that are distributed, multi-stable, and developing

破壊行為

ゴジラ