Building Interactive Systems

Evalaution & Presentation

Professor Bilge Mutlu | Spring 2023

What will we cover today?

→ Course evaluations

→ Basics of evaluation

→ Forms of evaluation

→ Types of data

→ Types of analyses

→ Basics of communication

→ Forms of communication

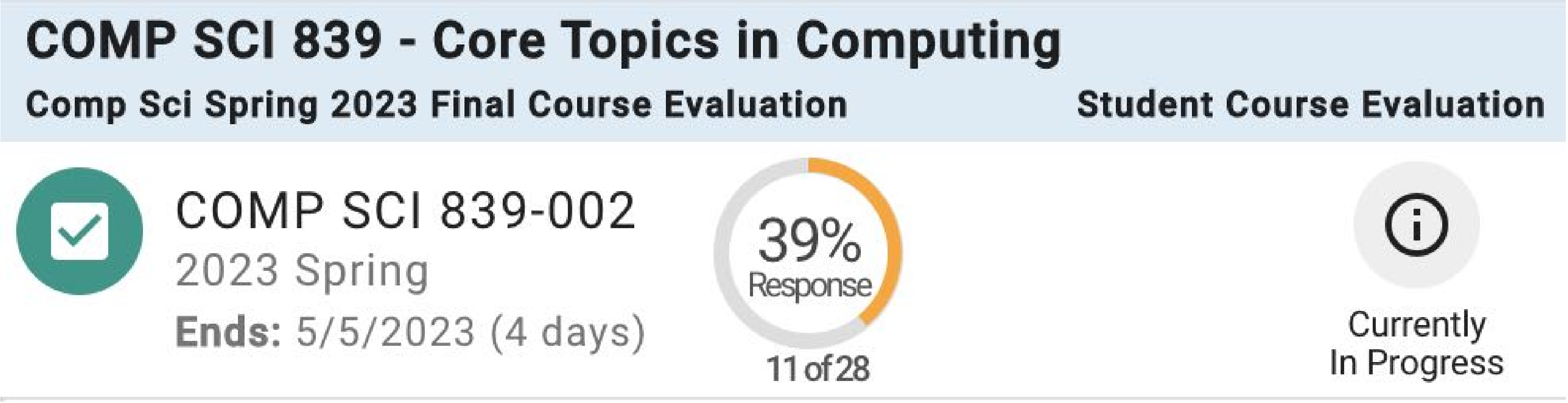
Course Evaluations

Please complete course evaluations.

→ Provide feedback on course scope, structure, mechanics

→ Address, *Should the department offer this course in the future?*

If we get to 90+ by Wednesday, I will bring treats 🍪🍩



Basics of Evaluation

Why evaluate?

→ To test claims we make in our research. For example, in our projects, we make the following claim (where Y is our contribution):

We enable/improve X using Y under circumstances Z.

→ To identify design/implementation deficiencies in our systems that we can improve.

What are we evaluating when we evaluate?

→ Test claims → testing the relationship between X and Y

→ Identify deficiencies → testing how well our systems works

Types of Evaluation

Evaluation (of systems research/development) can be **formative** or **summative**.[1]

Formative evaluation: Done to find and fix problems with an existing product in order to make it more usable.

The audience is the project team itself. Less formal.

Summative evaluation: Done to understand what is usable about a design after it is complete, or, what is working.

Broad audience. More formal.

[1] Formative vs. summative research

Types of Evaluation

Evaluation goals can include:

→ Hypothesis testing

→ Usability evaluation

→ System-level evaluation

→ The "user study"

→ System validation studies

Hypothesis Testing

Definition: A formal process of testing an assumption regarding a population parameter (e.g., the effects of Y on X).[2]

1. State (at least) two hypotheses so that only one can be right (null vs. alternative hypotheses).
2. Formulate a plan for testing these hypotheses.
3. Carry out the plan and analyze the sample.
4. Evaluate the hypotheses, rejecting or accepting the null hypothesis.

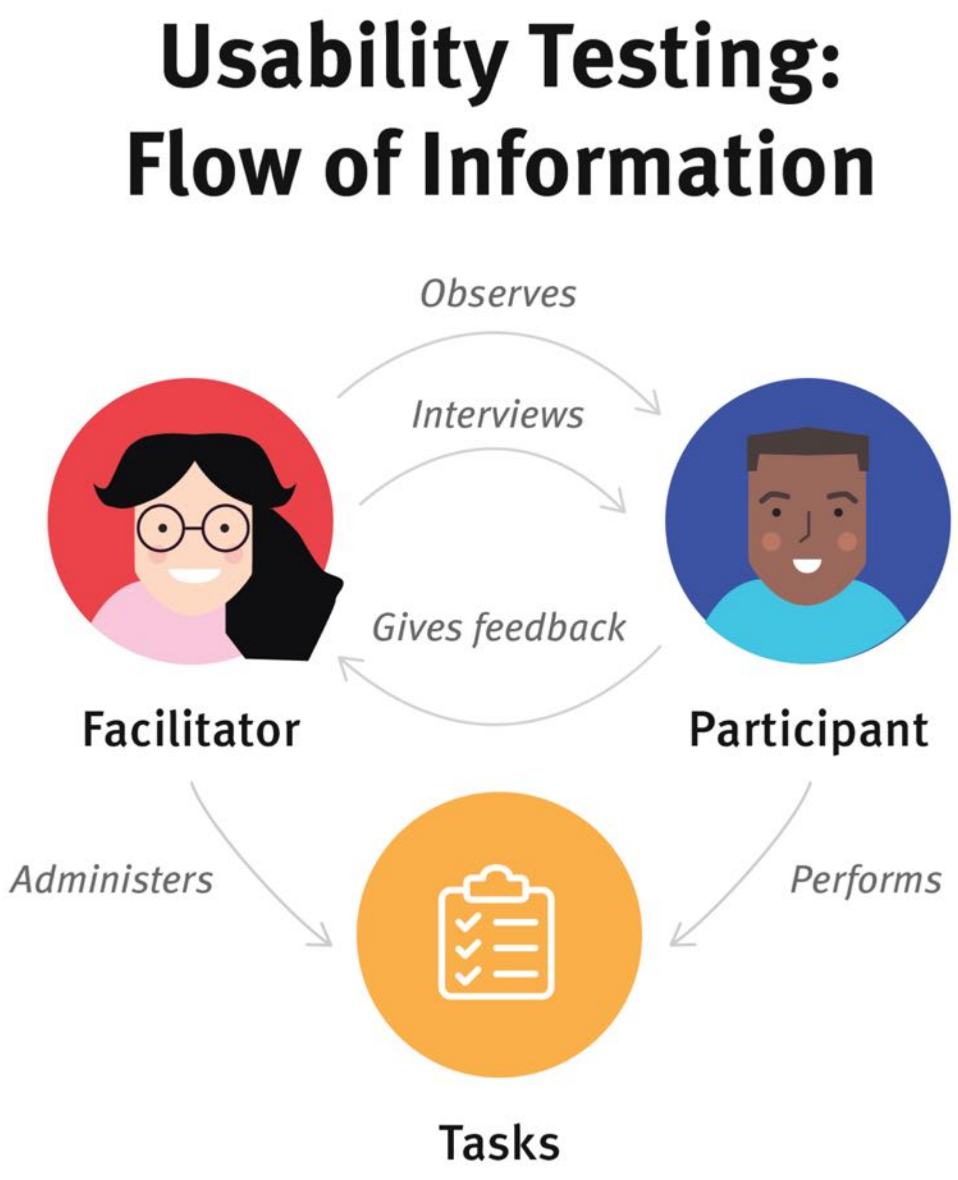
[2] Hypothesis to be tested

Usability Testing

Definition: A simulated use session with technology aimed to identify design problems, uncover opportunities to improve the system, and learn about the target user's behavior and preferences.[3]

1. Researcher asks a participant to perform tasks, usually using one or more specific user interfaces.

2. Researcher observes the participant's behavior and listens for feedback.



[3] Usability Testing 101

System-level Evaluation

Definition: A naturalistic form of evaluation that relies on multivariate statistics to determine the relationship between system features and desired interaction outcomes with the system.

$$Y_i = (b + 0 + b_1 x_{1i} + b_2 x_{2i} + \ldots + b_n x_{ni}) + i$$

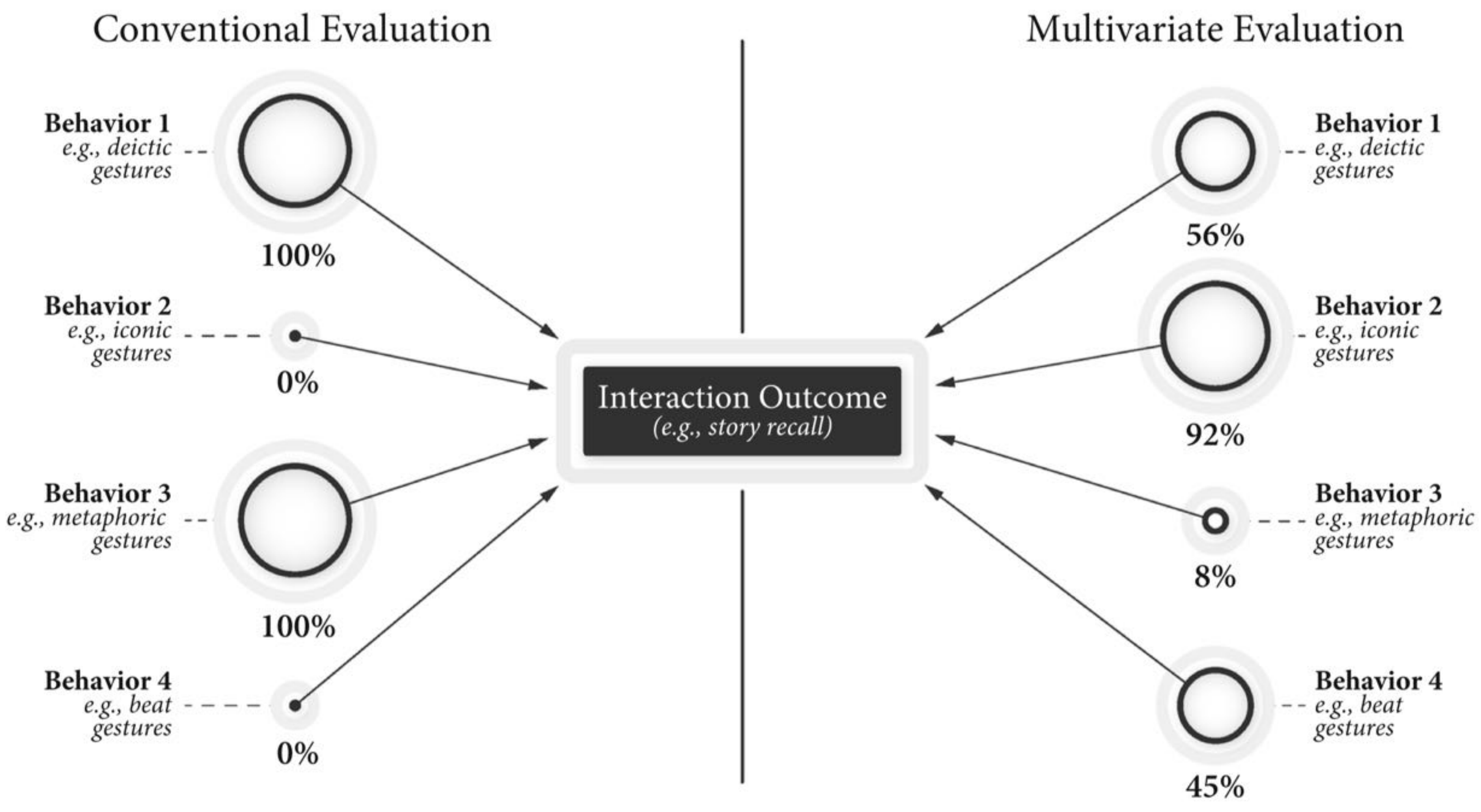
System-level Evaluation: *Example*[4]

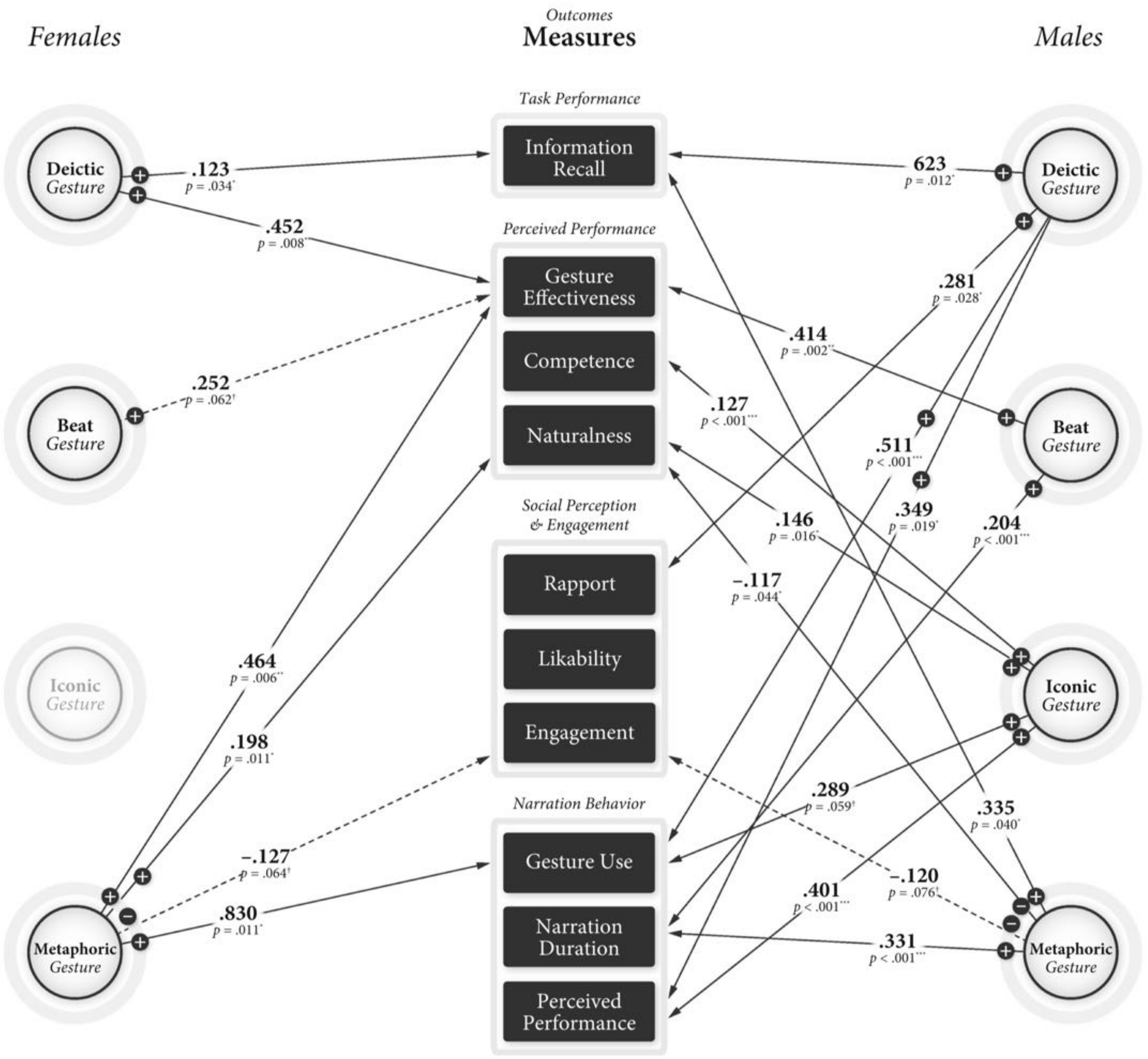
Abbreviation	Measure	Min	Max	Mean	Stdev
Time	Duration of interaction (in minutes)	5.10	21.14	11.07	4.05
UU/min	User utterances per minute	3.16	7.90	3.55	2.20
RU/min	Robot utterances per minute	5.44	15.19	10.91	2.50
Obj	# Objects learned	2	19	9.43	3.92
Obj_{corr}	# Correctly learned objects	1	17	8.79	3.85
ObjDemo	# Object demonstrations	2	25	10.07	5.97
Obj/min	Objects learned per minute	0.2	1.6	0.91	0.39
$Steps_{ObjDemo}$	Average interaction steps for object demo	4.23	6.80	5.80	0.60
$Steps_{ObjReq}$	Average interaction steps for object request	0.00	4.00	3.41	1.21
Gaps	Length of global pauses (in seconds)	0.53	3.32	1.50	0.62
Overlaps	% of time UU and RU overlapping	0%	49%	20%	14%
Repair	% of RU dealing with repair	4%	38%	14%	8%
RU_{inapp}	% of inappropriate RU	1%	26%	6%	6%
NPattern	# Completed interaction patterns	18	69	43.50	13.65
Ref	Success rate of reference resolution	36%	100%	79%	15%
Learn	Success rate of object learning	45|35.6%	100|79%	78|62%	20|15%
Check	Success rate of object check	0%	100%	53%	29%
$Check_{correct}$	Correctness rate of object check	0|0%	100|53%	55|29%	38|20%
UU_{ooc}	Out-of-capability UU	0	28	7.64	6.80
UU_{ooc} %	Percentage of out-of-capability UU	0%	21%	6%	5%

Measure	Function	R^2		β	Significance
Ease of use	-0.30 + 4.33(Obj/min) + 1.15(ObjDemo) + 0.12(Time)	.471	Obj	.792	p<.001
			ObjDmo	.466	p<.05
			Time	.399	p=.059
Efficiency	No significant model				
Clarity	4.63 - 6.92(Repair) + 1.39(Obj/min)	.367	Repair	-.467	p<.01
			Obj	.336	p<.05
Pleasantness	3.58 + 0.18(Learn)	.184	Learn	.465	p<.05
Understandability	3.78 + 0.14(Ref)	.196	Ref	.443	p<.05
Efficiency	1.42 - 5.09(Repair) + 0.61($Steps_{ObjDemo}$) - 0.06(UU_{ooc}) - 0.10($Steps_{ObjReq}$)	.441	Repair	-.403	p<.05
			$Steps_{ObjDemo}$	.394	p<.05
			UU_{ooc}	-.404	p<.05
			$Steps_{ObjReq}$	-.316	p=.081
Usability	4.03 - 0.22($Steps_{ObjDemo}$) + 2.39(Obj/min) - 0.21(UU/min) + 0.03(NPattern)	.553	$Steps_{ObjDemo}$	-.747	p<.001
			Obj	.736	p<.001
			UU/min	-.521	p<.01
			NPattern	.391	p<.05
Robustness	5.07 - 6.08(Repair)	.183	Repair	-.428	p<.05
Likeability	No significant model				
Cooperativeness	1.89 + 1.79(Obj/min) + 0.001(Gaps)	.446	Obj_{corr}	.698	p<.001
			Gaps	.418	p<.05
Perceived Intelligence	3.02 + 0.001(Gaps)	.171	Gaps	.171	p<.05
Animacy	3.67 + 1.09(Obj/min) - 0.118(UU/min)	.281	Obj	.443	p<.05
			UU/min	-.332	p=0.074

[4] Peltason et al. (2012). Talking with robots about objects: a system-level evaluation in HRI. *HRI 2012.*

System-level Evaluation: *Another Example*[5]





[5] Huang & Mutlu (2014). Multivariate evaluation of interactive robot systems. *Autonomous Robots*.

The "User Study"

Definition: Multipurpose, multi-method evaluation conducted to test claims about the effectiveness or use of new technology by the target user group.

→ Mixed methods, combining quantitative and qualitative metrics and analyses

→ Semi-formal, involving a modest number of participants with well-outlined procedures

→ Uses validated and exploratory metrics

→ May include the target population or a convenience sample

→ Most commonly used evaluation method used HCI systems research

Guerilla Testing Approach

Definition: a quick and inexpensive way of testing a prototype or website with real users. Instead of recruiting a specific targeted audience to take part in sessions, participants are approached in public places and asked to take part in research.[6] [7]

[6] What is Guerrilla Usability Testing?

[7] Guerilla Testing with Usability Cafe

System Validation Studies

Definition: Testing of system capabilities—component- or system-level—to assess system performance, error, etc.

→ Machine learning validation (e.g., cross-validation)

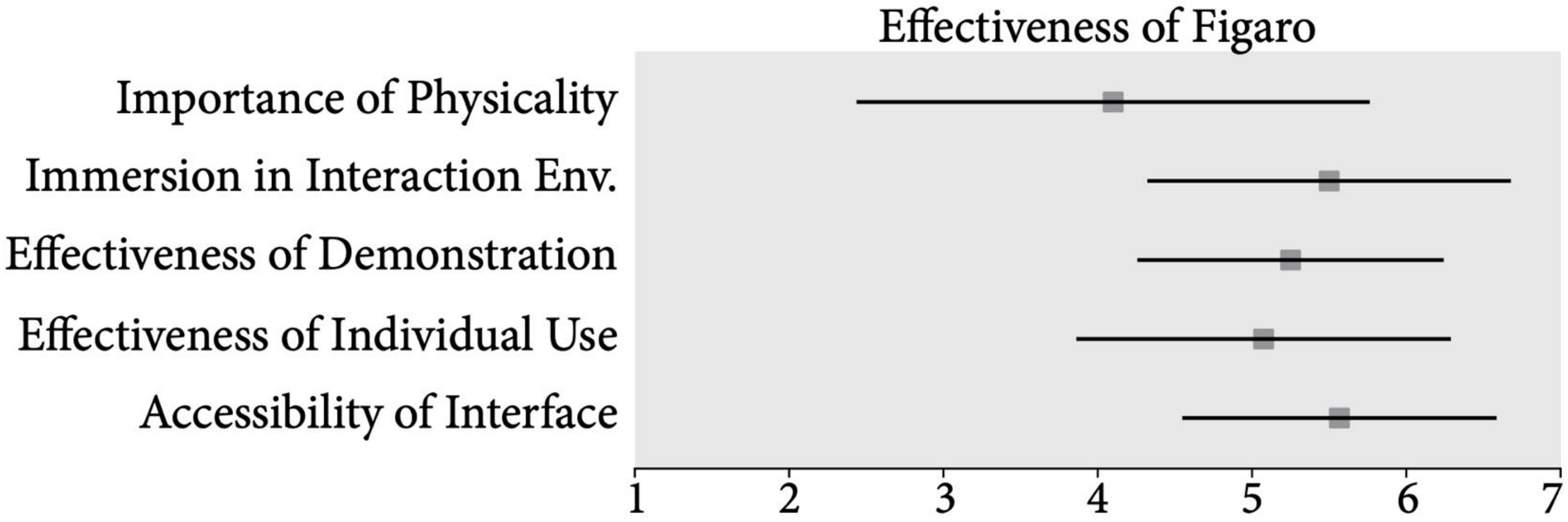
→ Simulated data experiments

→ Case study evaluation

Types of Data

Quantitative Methods

Definition: User study approaches that aim to *quantify* one or more dimension of user experience with the system, such as performance, usability, ease of learning, and satisfaction by collecting numerical data.[8]



[8] Porfirio et al. (2021). Figaro: A tabletop authoring environment for human-robot interaction. *CHI 2021.*

Qualitative Methods

Definition: User study approaches that aim to understand user motivations and use patterns with a system through observation and interviews. Qualitative data can also be quantified.[8]

Theme 1: Expressing ideas tangibly
Um, pointing is basically without saying you can direct [...] the human where to go. It's answering by doing rather than by speaking. So, I think it's [...] very expressive. (P9.34)

[8] Porfirio et al. (2021). Figaro: A tabletop authoring environment for human-robot interaction. *CHI 2021.*

Combining Insights[9]

- → Quantitative methods address "to what extent" questions.
- → Qualitative methods address "how" (e.g., process) and "why" (e.g., motivation) questions.
- → We often need to address both and thus conduct *mixed-methods* user studies.

[9] Quantitative vs. Qualitative Usability Testing

Types of Data

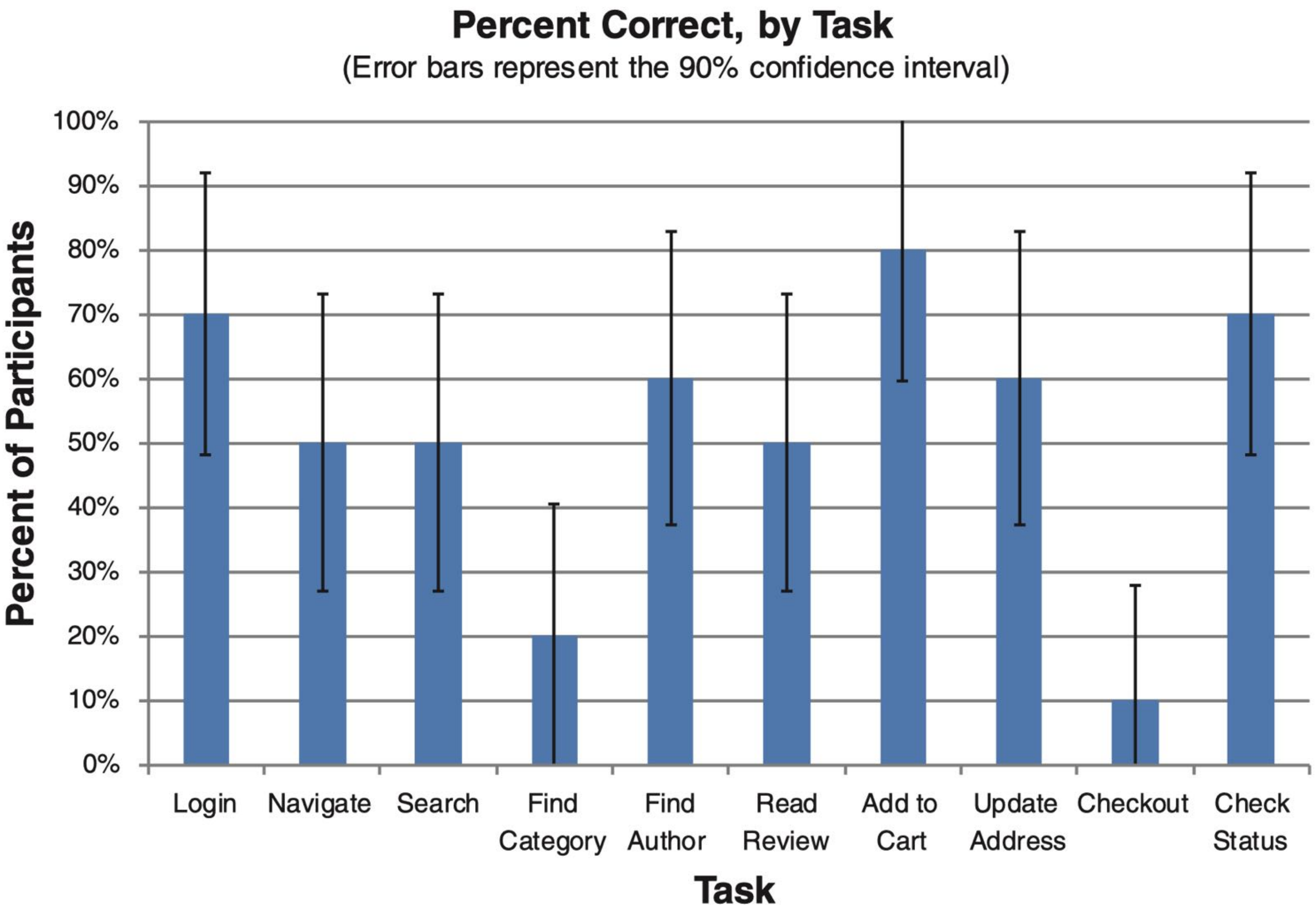
- Quantitative methods:
  - **Objective data** (e.g., task performance, error rate)
  - Behavioral/physiological data (e.g., gaze fixations, heart rate)
  - **Self-reported data** (e.g., SUS scale)
  - Issue-based data (e.g., number of task failures from system logs)
- Qualitative methods:
  - **Interview data** (e.g., responses to open-ended questions)
  - User observations (e.g., user workflows)

Working with objective data

Types of objective data:[10]

1. **Task success** — how effectively users are able to complete a given set of tasks (e.g., binary success, levels of success)
2. **Time-on-task** — how much time is required to complete a task
3. **Errors** — the mistakes made during a task
4. **Efficiency** — the amount of effort a user expends to complete a task (e.g., the number of button presses on a mobile app)
5. **Learnability** — how performance changes over time

[10] Albert & Tullis (2022). *Measuring the User Experience: Collecting, Analyzing, and Presenting UX Metrics*. Morgan Kaufmann.



[10] Albert & Tullis (2022). *Measuring the User Experience: Collecting, Analyzing, and Presenting UX Metrics*. Morgan Kaufmann.

Considerations in collecting and using objective data

→ Measured manually or automatically

→ Objective, no validation is necessary

→ For task success, criteria must be set

→ Most powerful type of data in systems research

→ Can be noisy and highly variable across individuals

Working with self-reported data

User experience dimensions:[11]

→ **Usability** — the level of ease at which users can perform tasks supported by the system

→ **Usefulness** — the utility of the system in supporting user goals

→ **Desirability** — the user interest, motivation, and satisfaction in using the system

Commonly used metrics:

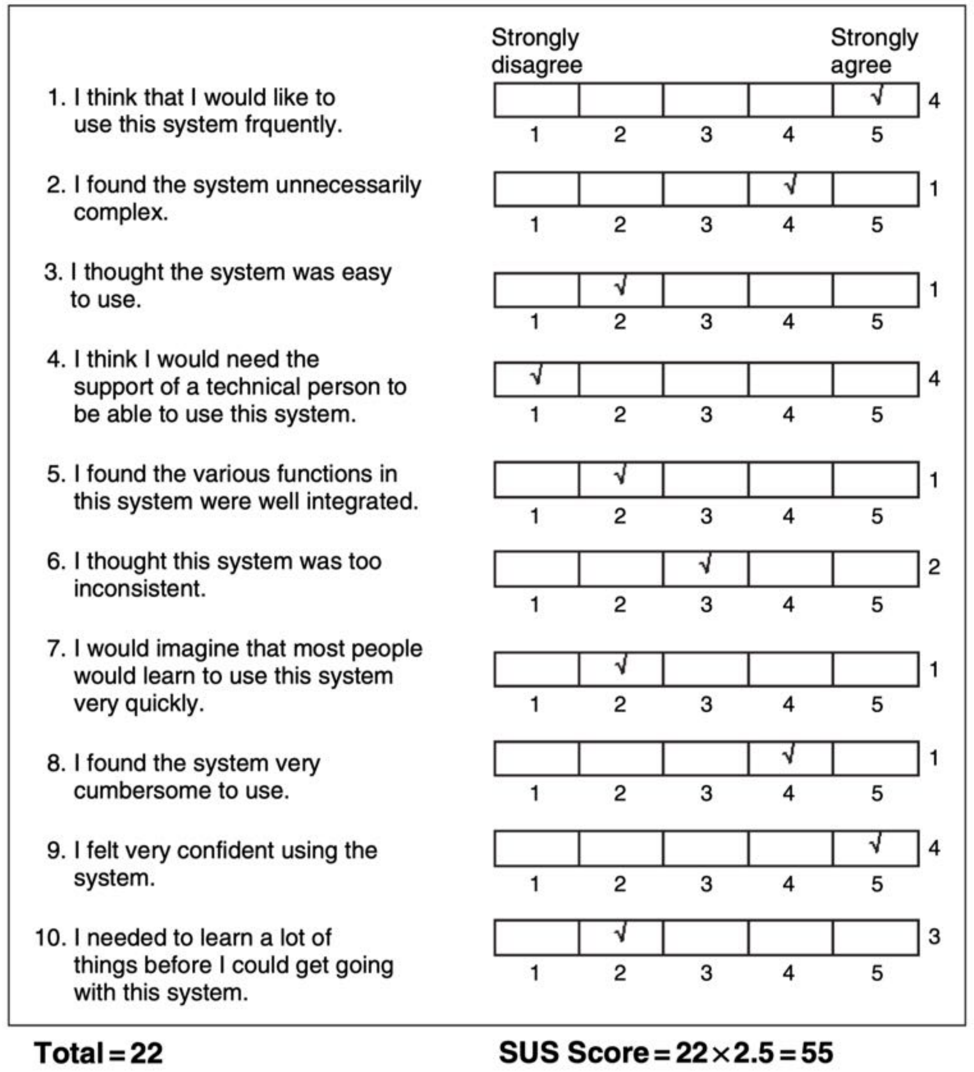
→ System usability scale (SUS)

[11] Usability vs Desirability in Mobile UX

SUS[12] [13]

Ten-item questionnaire that focuses on usability.

Can be used for relative comparison or absolute benchmarking.



	Strongly disagree 1	2	3	4	Strongly agree 5	
1. I think that I would like to use this system frquently.					√	4
2. I found the system unnecessarily complex.				√		1
3. I thought the system was easy to use.		√				1
4. I think I would need the support of a technical person to be able to use this system.	√					4
5. I found the various functions in this system were well integrated.		√				1
6. I thought this system was too inconsistent.			√			2
7. I would imagine that most people would learn to use this system very quickly.		√				1
8. I found the system very cumbersome to use.				√		1
9. I felt very confident using the system.					√	4
10. I needed to learn a lot of things before I could get going with this system.		√				3

Total = 22 **SUS Score = 22 × 2.5 = 55**

[12] How to use the SUS

[13] Image source: Albert & Tullis, 2013, Measuring the User Experience

USE[14]

Includes four sub-scales for:

- → usefulness
- → ease of use
- → ease of learning
- → satisfaction

Items in the USE Questionnaire with Labels in Current Study

Label	Content
UU1	It helps me be more effective.
UU2	It helps me be more productive.
UU3	It is useful.
UU4	It gives me more control over the activities in my life.
UU5	It makes the things I want to accomplish easier to get done.
UU6	It saves me time when I use it.
UU7	It meets my needs.
UU8	It does everything I would expect it to do.
UE1	It is easy to use.
UE2	It is simple to use.
UE3	It is user friendly.
UE4	It requires the fewest steps possible to accom-plish what I want to do with it.
UE5	It is flexible.
UE6	Using it is effortless.
UE7	I can use it without written instructions.
UE8	I don't notice any inconsistencies as I use it.
UE9	Both occasional and regular users would like it.
UE10	I can recover from mistakes quickly and easily.
UE11	I can use it successfully every time.
UL1	I learned to use it quickly.
UL2	I easily remember how to use it.
UL3	It is easy to learn to use it.
UL4	I quickly became skillful with it.
US1	I am satisfied with it.
US2	I would recommend it to a friend.
US3	It is fun to use.
US4	It works the way I want it to work.
US5	It is wonderful.
US6	I feel I need to have it.
US7	It is pleasant to use.

Note. Labels were not used in the actual online survey. UU = Usefulness, UE = Ease of Use, UL = Ease of Learning, US = Satisfaction.

[14] Gao et al. (2018). Psychometric evaluation of the use (usefulness, satisfaction, and ease of use) questionnaire for reliability and validity. *HFES 2018.*

Considerations in collecting and using subjective data[10]

- → Administered using rating/Likert scales
  - → Rating scale:
    Rate between 1 and 5 where 1 is least and 5 is most
  - → Likert scale:
    Rate between strongly disagree and strongly agree
  - → Semantic differential scale
    Weak ○ ○ ○ ○ ○ Strong
- → Can be captured on paper, orally, or through a computer/online form

[10] Albert & Tullis (2022). *Measuring the User Experience: Collecting, Analyzing, and Presenting UX Metrics*. Morgan Kaufmann.

Working with qualitative data

Generally in text form, qualitative data can include:

→ Fieldnotes from observations (that the researcher generates)

→ Transcripts from structured or semi-structured interviews

→ Archival narrative data (e.g., Twitter posts, Reddit discussions)

Considerations in collecting and using interview data

→ Design a small number of questions that get to aspects of user experience, motivation, and perceptions of the system

→ Ask *what*, *how*, *why* questions; avoid questions that can be answered with "yes" or "no"

→ Ask probing questions (e.g., "Can you tell me a bit more?") and follow up questions (e.g., "Why do you say that?")

Considerations in collecting and using interview data

Example questions:[15]

- → How often do you read?
- → What systems do you have in place to encourage your child to read?
- → If you had a technology that worked with you while you read, what would it do?

[15] Michaelis & Mutlu (2017). Someone to read with: Design of and experiences with an in-home learning companion robot for reading. *CHI 2017.*

Data Analysis Methods

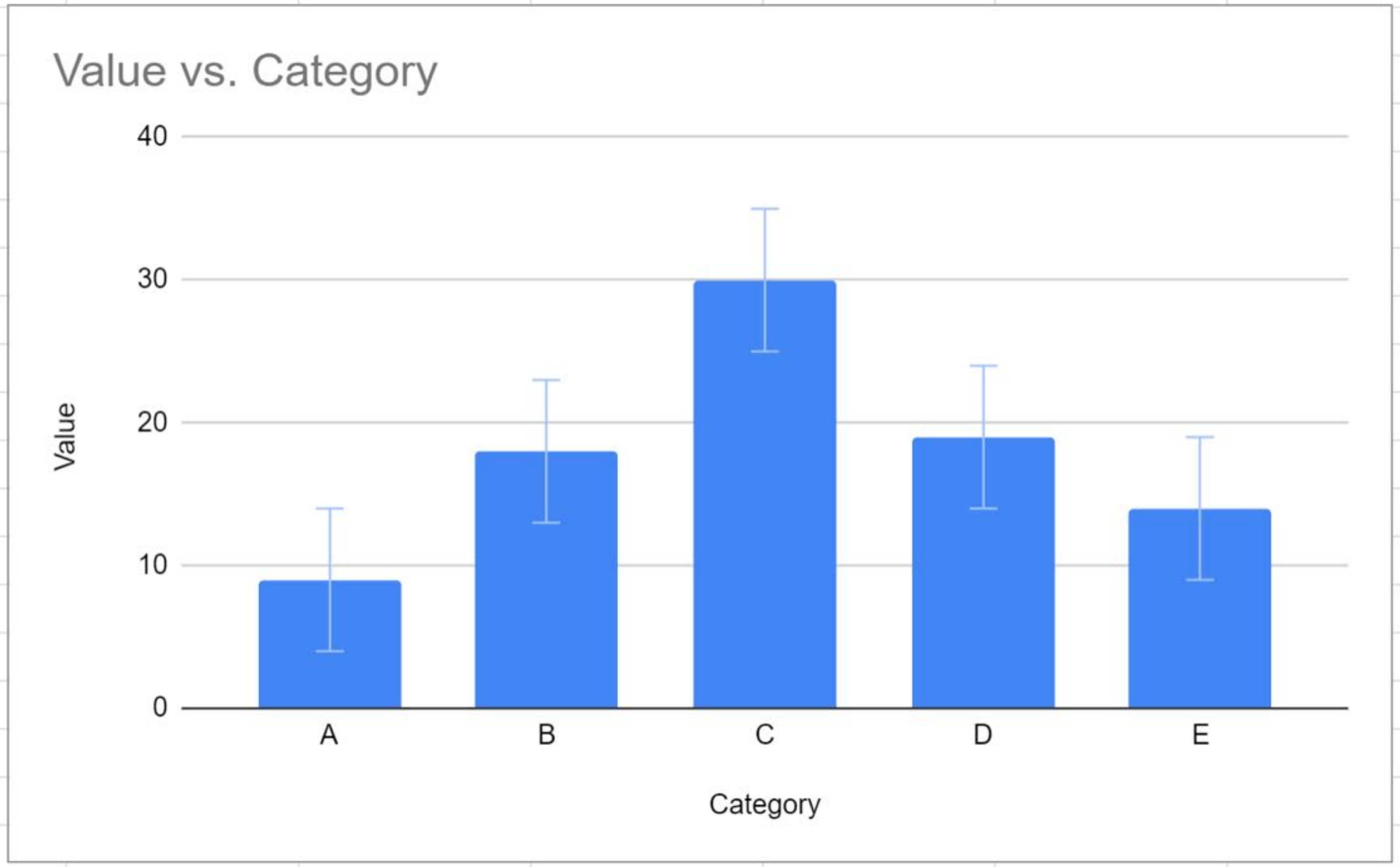
Quantitative Analysis Methods[16]

- **Descriptive statistics:** used to describe the *distributional* characteristics or features of a dataset
  - *Central tendency:* mean, median, mode
  - *Variability:* standard deviation, min/max, range, kurtosis, skewness
- **Inferential statistics:** used to make generalizations about a larger population based on a representative sample of that population
  - Used in hypothesis testing, or comparative user studies
  - Used to capture causal and correlational effects

[16] What’s the Difference Between Descriptive and Inferential Statistics?

17

	A	B	C	D	E
1	**Dataset**		**Mean**	16.3	=AVERAGE(A2:A21)
2	5		**Median**	14.5	=MEDIAN(A2:A21)
3	20		**Mode**	13	=MODE(A2:A21)
4	14		**Range**	31	=MAX(A2:A21) - MIN(A2:A21)
5	13		**Standard Dev.**	9.0618	=STDEV(A2:A21)
6	8		**Sample size**	20	=COUNTA(A2:A21)
7	6				
8	6				
9	15				
10	10				
11	12				
12	17				
13	27				
14	22				
15	21				
16	13				
17	29				
18	4				
19	18				
20	31				
21	35				
22					
23					
24					



[17] How to Calculate Descriptive Statistics in Google Sheets, How to Add Error Bars to Charts in Google Sheets

Qualitative Analysis Methods

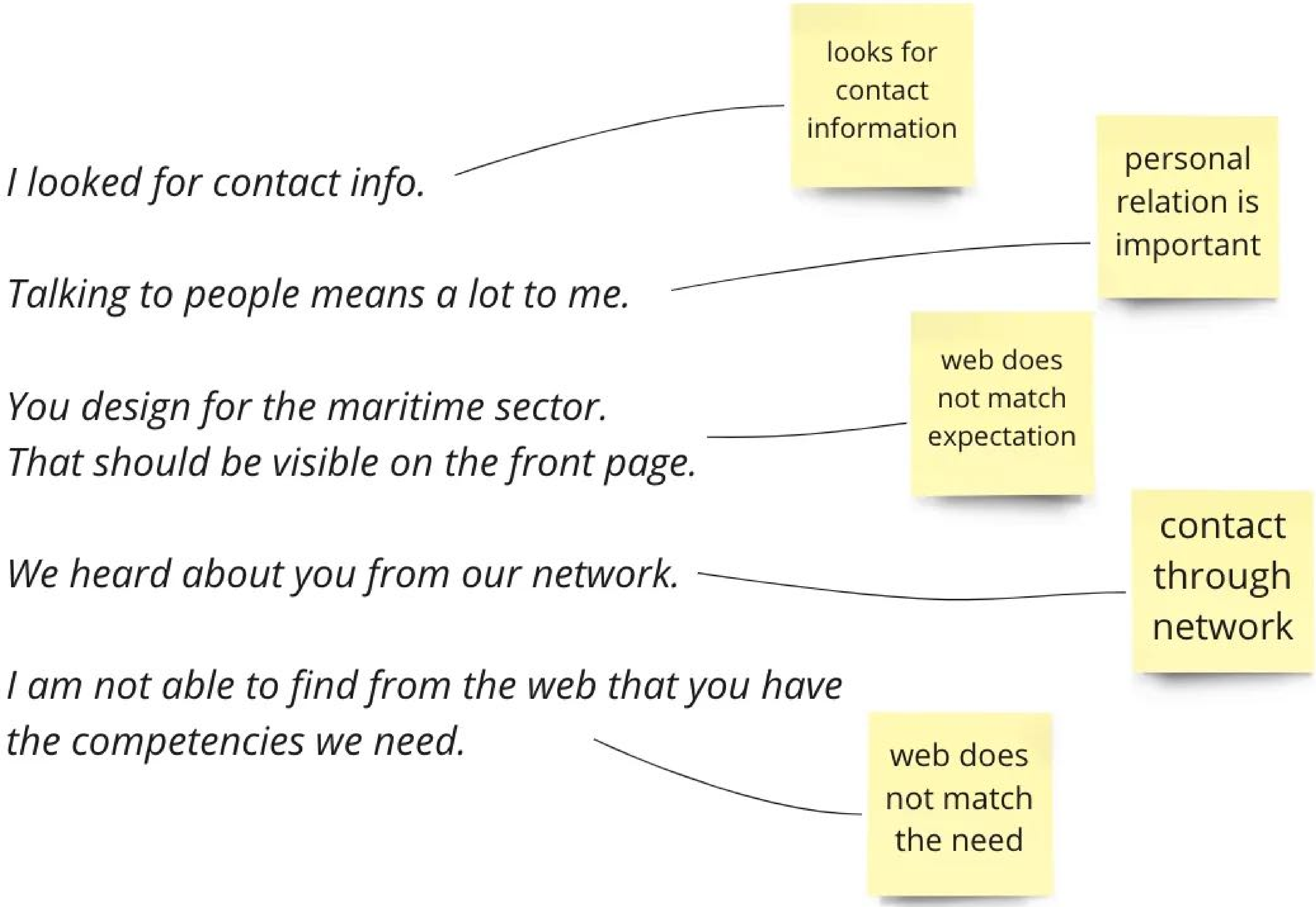
The goal of qualitative analysis is extract insight from rich, complex textual data.

Um, pointing is basically without saying you can direct [...] the human where to go. It's answering by doing rather than by speaking. So, I think it's [...] very expressive. (P9.34)[8]

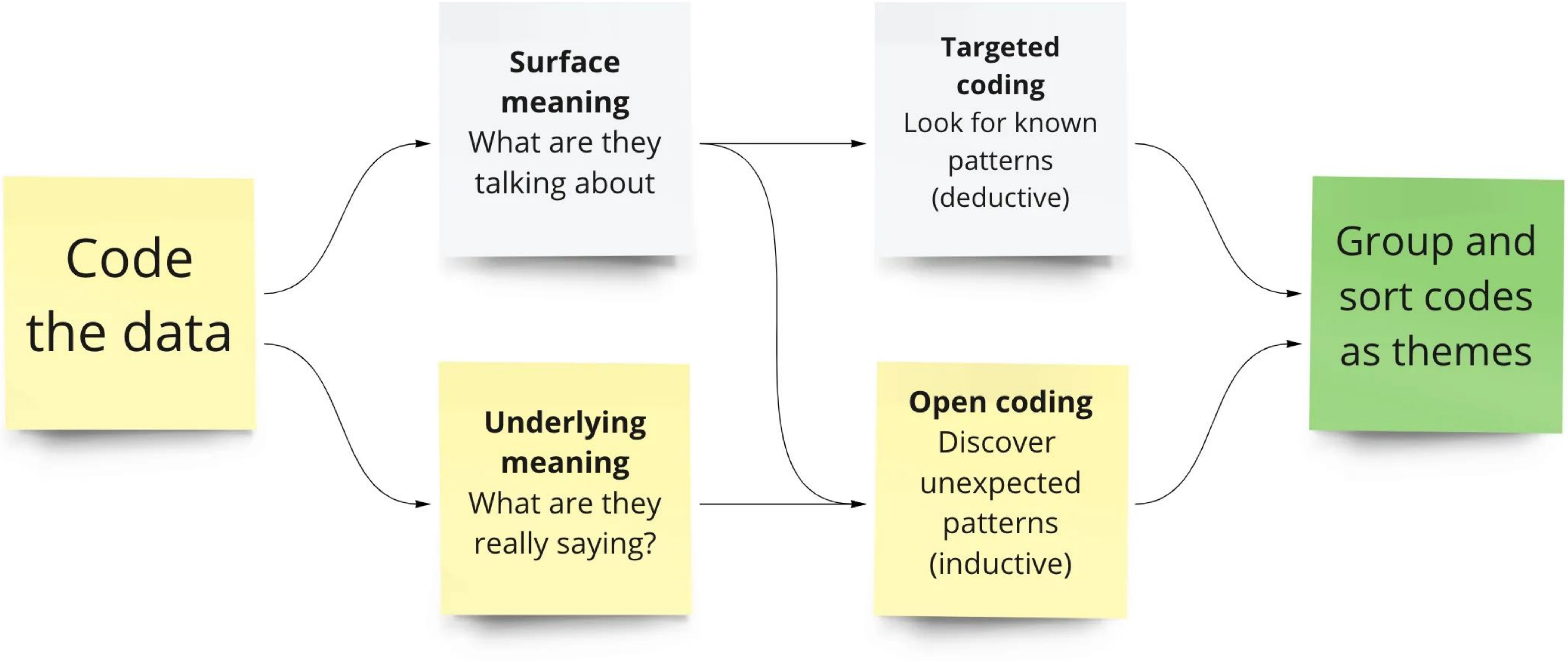
→ *Expressing ideas tangibly*

[8] Porfirio et al. (2021). Figaro: A tabletop authoring environment for human-robot interaction. *CHI 2021.*

18

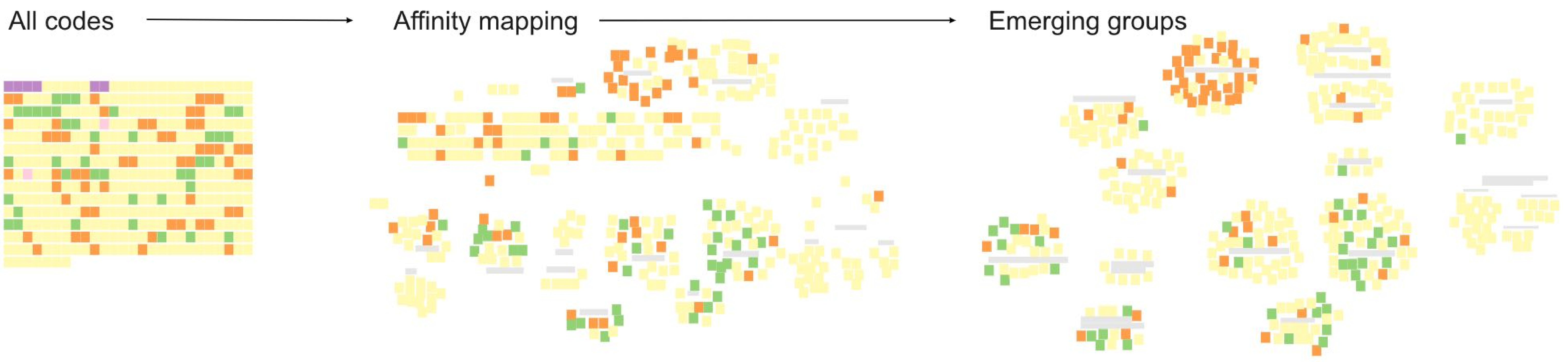


[18] Insight that matters – how to analyse qualitative data in design



[18] Insight that matters – how to analyse qualitative data in design

18



[18] Insight that matters – how to analyse qualitative data in design

Users in Evaluation

Who are the users in a user study?

→ User studies involve asking users to use your system and gathering data.

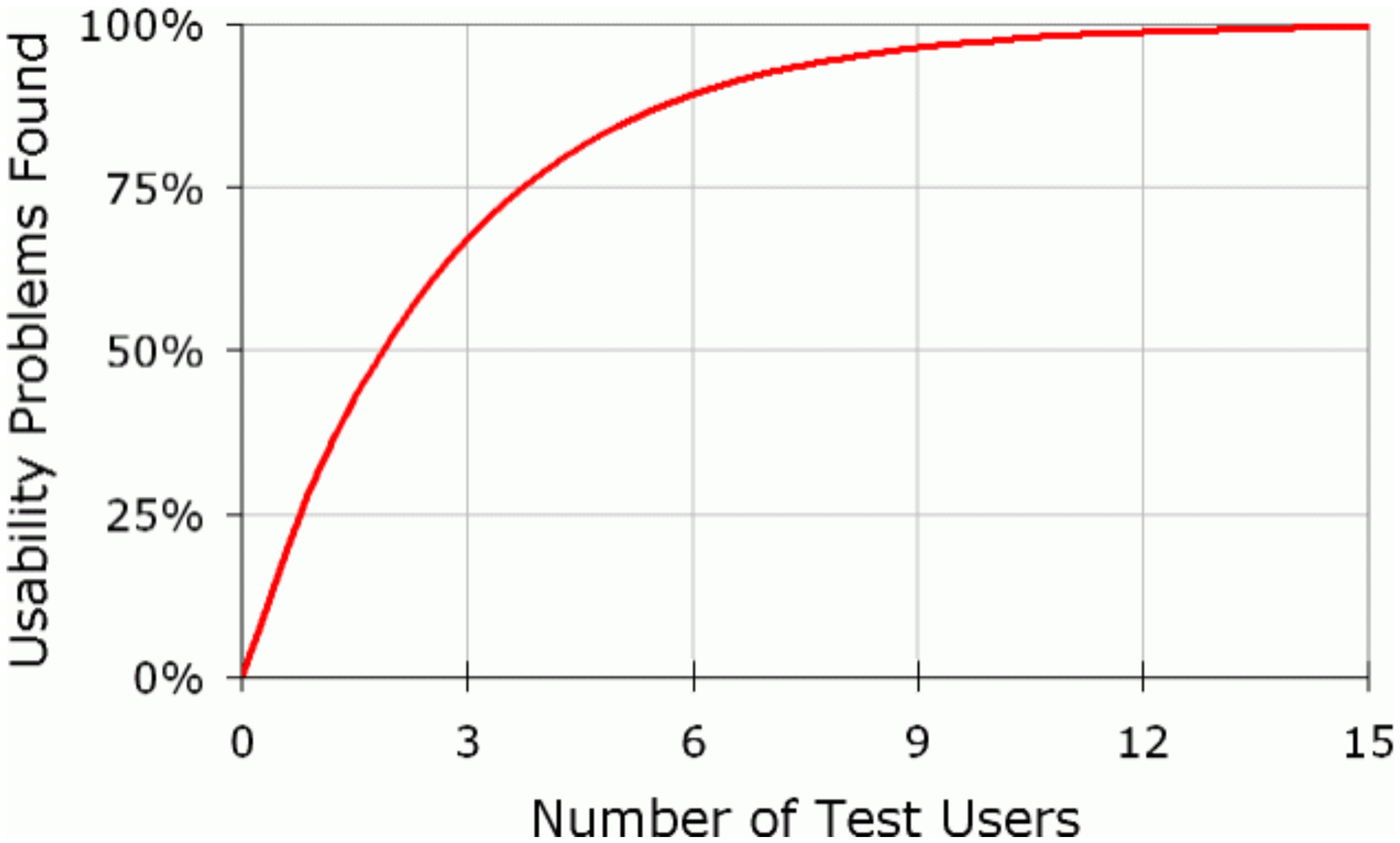
→ These users are called "study participants"

→ Ideally, representatives of the target user of population of your system.

→ Often working with participants who approximate your target population is OK.

How many participants do I need?

- In usability testing, 5 participants is considered sufficient[19]
- In user studies, the rule of thumb is:
  - A minimum of 8 participants for each comparative category
  - A minimum of 16 participants for noncooperative designs



[19] Why You Only Need to Test with 5 Users

User Study Checklist

User Study Checklist

- → What is the systems claim?
- → What facet of the claim will be tested?
- → What evaluation approach should be followed?
- → What metrics should be used?
- → Who are the participants?
- → How many participants should be recruited?
- → How will the study be administered?
- → How will data be captured?
- → How will data be analyzed?

Presentation

Why do we have to present?[20]

Science advances through communication and building on other people's work.

Publication: the action or process of making public[21]

[20] On the Shoulders of Giants

[21] The History of Scientific Publishing

History of scientific communication[21]

- **17th Century:** Public sharing of scientific findings was in the form of oral transactions at meetings; publication was slow, expensive
- **1665:** Publication — periodicals started with the creation of *Philosophical Transactions* and *Journal des Scavanes* were created (1665)
- **1752:** Peer review — Royal Society assuming management of *Philosophical Transactions*
- **Late 19th Century:** New printing technologies, more widespread transportation, cost of paper, literacy rates facilitated the widespread readership of periodicals
- **Mid-late 20th Century:** Preprints became commonplace to rapidly share findings (arXiv was created in 1991)

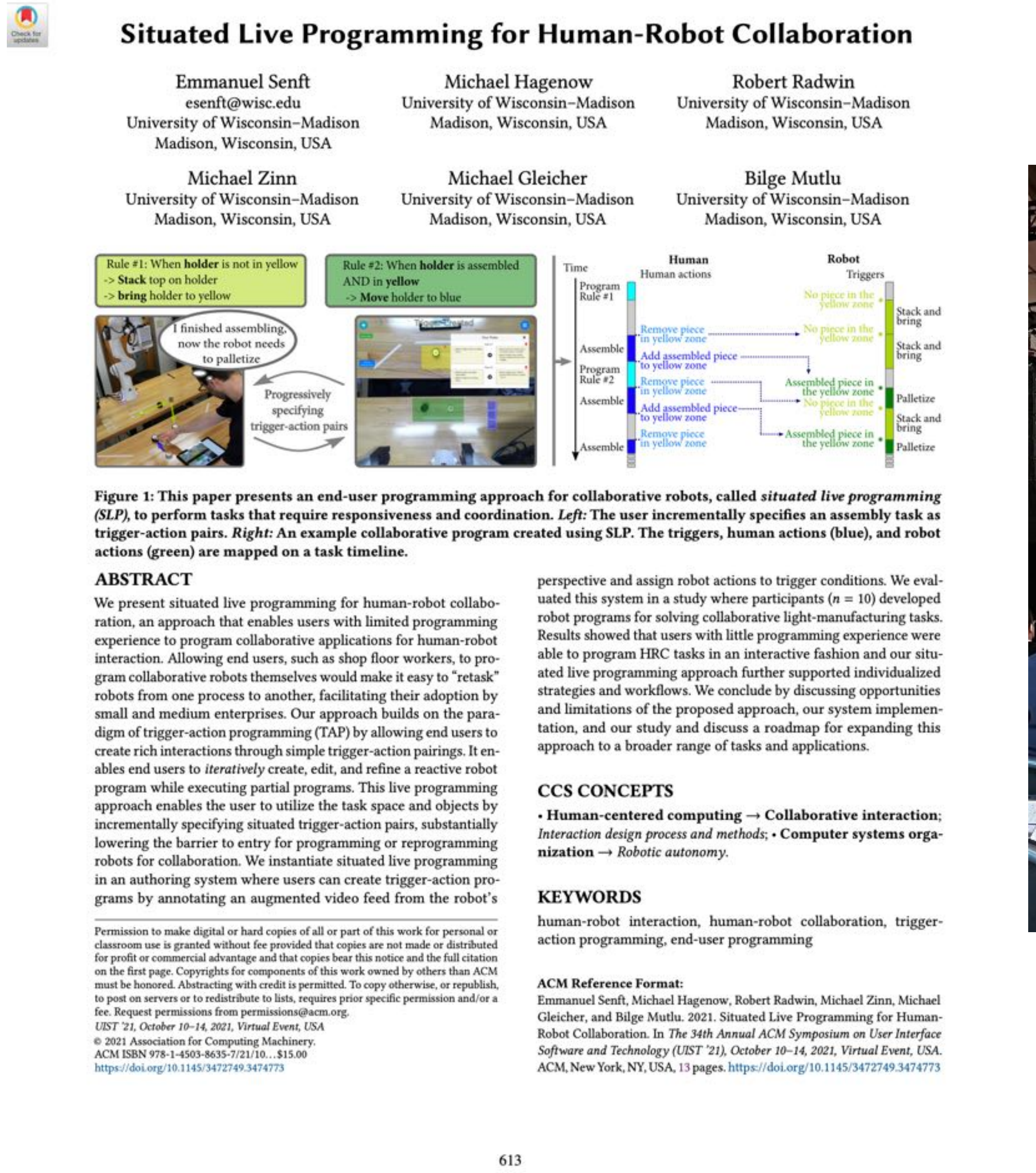
[21] The History of Scientific Publishing

Forms of communication

Forms of communication

Three points of contact with other scientists:

1. Publications in proceedings, journals
2. Oral communication at meetings
3. Multimedia content over the internet



Situated Live Programming for Human-Robot Collaboration

Emmanuel Senft
esenft@wisc.edu
University of Wisconsin–Madison
Madison, Wisconsin, USA

Michael Hagenow
University of Wisconsin–Madison
Madison, Wisconsin, USA

Robert Radwin
University of Wisconsin–Madison
Madison, Wisconsin, USA

Michael Zinn
University of Wisconsin–Madison
Madison, Wisconsin, USA

Michael Gleicher
University of Wisconsin–Madison
Madison, Wisconsin, USA

Bilge Mutlu
University of Wisconsin–Madison
Madison, Wisconsin, USA

Figure 1: This paper presents an end-user programming approach for collaborative robots, called *situated live programming (SLP)*, to perform tasks that require responsiveness and coordination. *Left:* The user incrementally specifies an assembly task as trigger-action pairs. *Right:* An example collaborative program created using SLP. The triggers, human actions (blue), and robot actions (green) are mapped on a task timeline.

ABSTRACT

We present situated live programming for human-robot collaboration, an approach that enables users with limited programming experience to program collaborative applications for human-robot interaction. Allowing end users, such as shop floor workers, to program collaborative robots themselves would make it easy to "retask" robots from one process to another, facilitating their adoption by small and medium enterprises. Our approach builds on the paradigm of trigger-action programming (TAP) by allowing end users to create rich interactions through simple trigger-action pairings. It enables end users to *iteratively* create, edit, and refine a reactive robot program while executing partial programs. This live programming approach enables the user to utilize the task space and objects by incrementally specifying situated trigger-action pairs, substantially lowering the barrier to entry for programming or reprogramming robots for collaboration. We instantiate situated live programming in an authoring system where users can create trigger-action programs by annotating an augmented video feed from the robot's perspective and assign robot actions to trigger conditions. We evaluated this system in a study where participants ($n = 10$) developed robot programs for solving collaborative light-manufacturing tasks. Results showed that users with little programming experience were able to program HRC tasks in an interactive fashion and our situated live programming approach further supported individualized strategies and workflows. We conclude by discussing opportunities and limitations of the proposed approach, our system implementation, and our study and discuss a roadmap for expanding this approach to a broader range of tasks and applications.

CCS CONCEPTS

• **Human-centered computing → Collaborative interaction**; *Interaction design process and methods*; • **Computer systems organization →** *Robotic autonomy.*

KEYWORDS

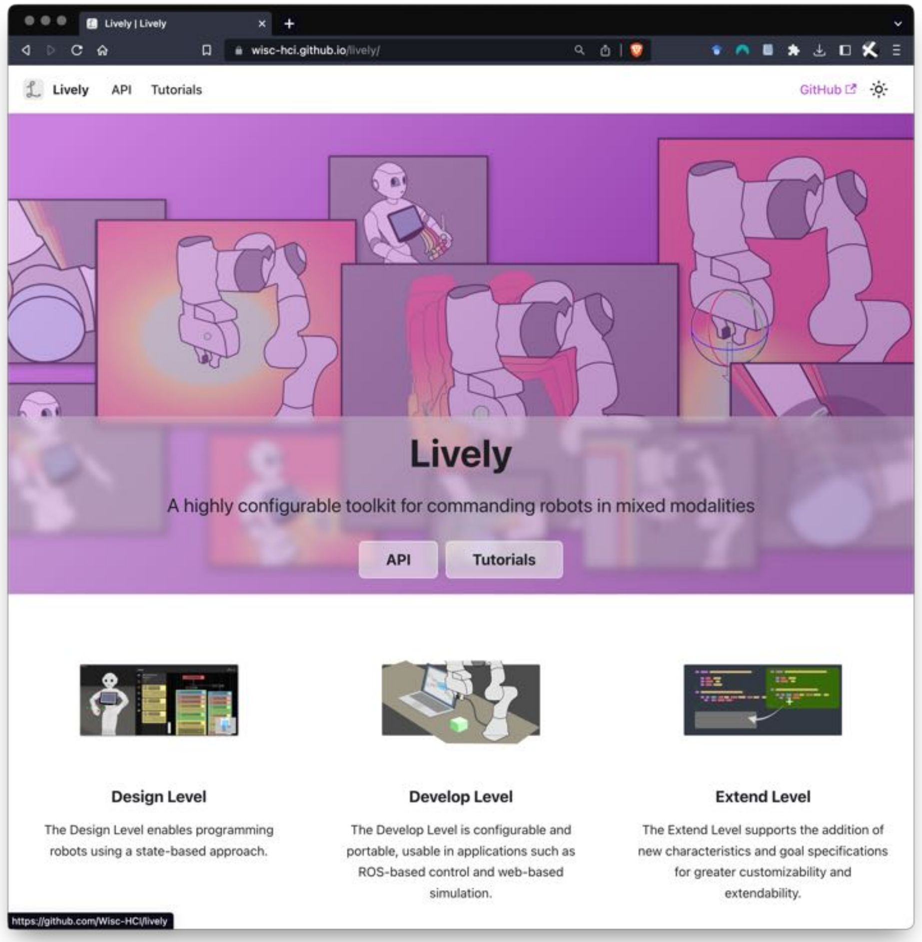
human-robot interaction, human-robot collaboration, trigger-action programming, end-user programming

ACM Reference Format:
Emmanuel Senft, Michael Hagenow, Robert Radwin, Michael Zinn, Michael Gleicher, and Bilge Mutlu. 2021. Situated Live Programming for Human-Robot Collaboration. In *The 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21), October 10–14, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3472749.3474773

UIST '21, October 10–14, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8635-7/21/10…$15.00
https://doi.org/10.1145/3472749.3474773

613



[22] Image sources: left, center, right

Papers

→ Format, length, community practices depend on the field, venue, publisher

→ Generally 4-12 pages (e.g., 4-page Science/Nature paper vs. 12-page proceedings)

→ Narrative presentation, supported by illustrations, data figures, tables, and supplementary material

→ Common flow: rhetoric → exposition → reflection

Situated Live Programming for Human-Robot Collaboration

Emmanuel Senft, esenft@wisc.edu, University of Wisconsin–Madison, Madison, Wisconsin, USA
Michael Hagenow, University of Wisconsin–Madison, Madison, Wisconsin, USA
Robert Radwin, University of Wisconsin–Madison, Madison, Wisconsin, USA
Michael Zinn, University of Wisconsin–Madison, Madison, Wisconsin, USA
Michael Gleicher, University of Wisconsin–Madison, Madison, Wisconsin, USA
Bilge Mutlu, University of Wisconsin–Madison, Madison, Wisconsin, USA

Figure 1: This paper presents an end-user programming approach for collaborative robots, called *situated live programming (SLP)*, to perform tasks that require responsiveness and coordination. *Left:* The user incrementally specifies an assembly task as trigger-action pairs. *Right:* An example collaborative program created using SLP. The triggers, human actions (blue), and robot actions (green) are mapped on a task timeline.

ABSTRACT

We present situated live programming for human-robot collaboration, an approach that enables users with limited programming experience to program collaborative applications for human-robot interaction. Allowing end users, such as shop floor workers, to program collaborative robots themselves would make it easy to "retask" robots from one process to another, facilitating their adoption by small and medium enterprises. Our approach builds on the paradigm of trigger-action programming (TAP) by allowing end users to create rich interactions through simple trigger-action pairings. It enables end users to *iteratively* create, edit, and refine a reactive robot program while executing partial programs. This live programming approach enables the user to utilize the task space and objects by incrementally specifying situated trigger-action pairs, substantially lowering the barrier to entry for programming or reprogramming robots for collaboration. We instantiate situated live programming in an authoring system where users can create trigger-action programs by annotating an augmented video feed from the robot's perspective and assign robot actions to trigger conditions. We evaluated this system in a study where participants ($n = 10$) developed robot programs for solving collaborative light-manufacturing tasks. Results showed that users with little programming experience were able to program HRC tasks in an interactive fashion and our situated live programming approach further supported individualized strategies and workflows. We conclude by discussing opportunities and limitations of the proposed approach, our system implementation, and our study and discuss a roadmap for expanding this approach to a broader range of tasks and applications.

CCS CONCEPTS

• **Human-centered computing → Collaborative interaction**; *Interaction design process and methods*; • **Computer systems organization → *Robotic autonomy*.**

KEYWORDS

human-robot interaction, human-robot collaboration, trigger-action programming, end-user programming

ACM Reference Format:
Emmanuel Senft, Michael Hagenow, Robert Radwin, Michael Zinn, Michael Gleicher, and Bilge Mutlu. 2021. Situated Live Programming for Human-Robot Collaboration. In *The 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21), October 10–14, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3472749.3474773

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
UIST '21, October 10–14, 2021, Virtual Event, USA
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8635-7/21/10...$15.00
https://doi.org/10.1145/3472749.3474773

613

Key challenges & competencies

- Clear articulation of core premise, central hypothesis, and/or the thesis of the paper
  - Research insight
- Persuasive storytelling of the *why* of the work — motivation, gap, promise
  - Narrative storytelling
- Rigorous technical writing of the *how* of the work — what was done, how it was done
  - Technical writing
- Insightful reflection on the *what* of the work — findings, implications, and limitations

Presentations[23]

→ Common formats: oral, poster, or demo presentation

→ **Oral presentation**

  → 5–20 minute talk, accompanied by projected slides

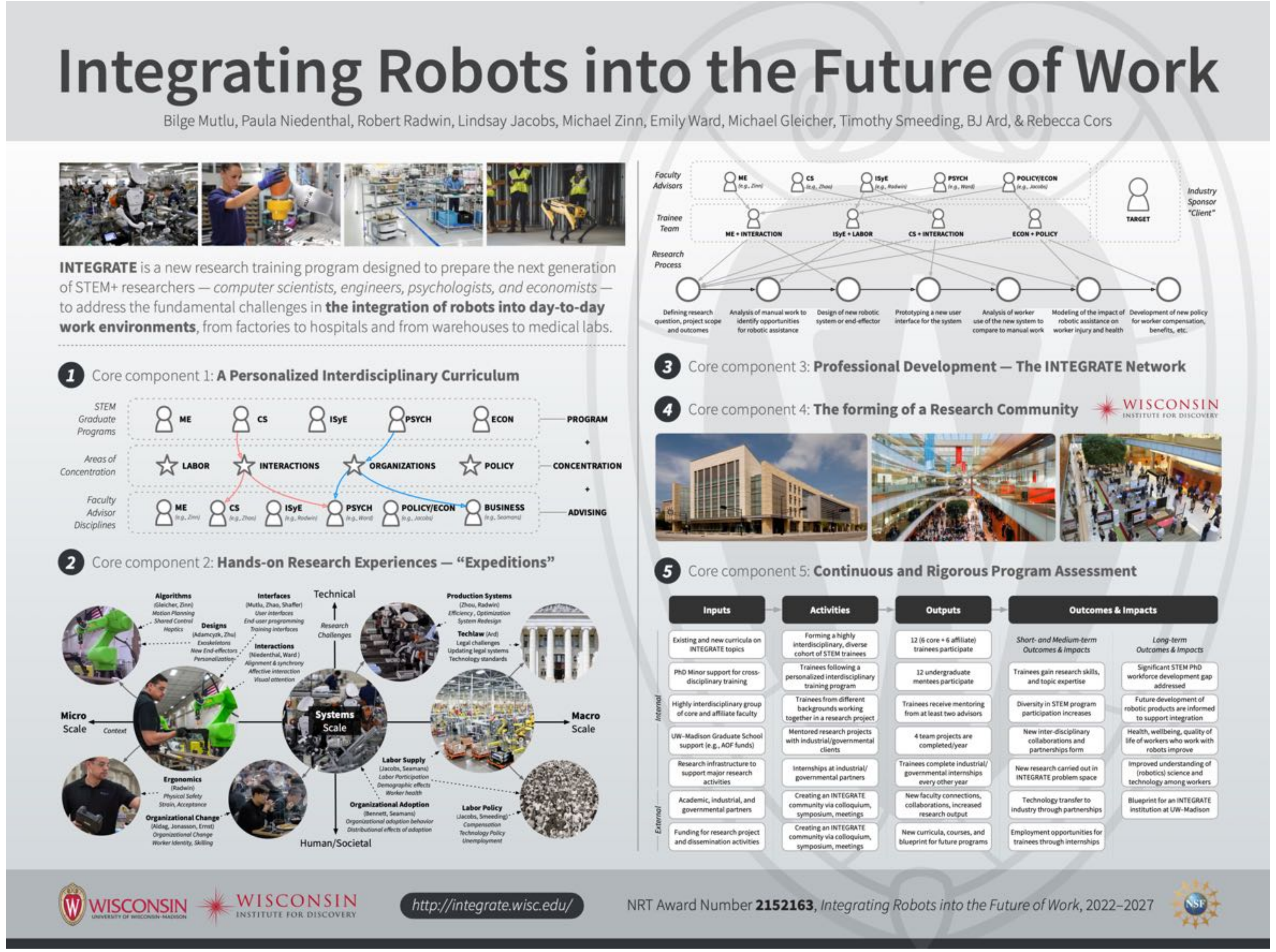
  → An invitation for the audience to read your paper

  → Can follow the same flow: rhetoric → exposition → reflection

[23] Image source

→ **Poster presentation**

- → A0 or 32x24 in horizontal or vertical poster
- → Assume that you will be there
- → Primarily visual with little text
- → Add letter-sized printouts, business cards
- → Add QR code for a digital version



→ **Demo presentation**[24]

→ Usually open-ended with no guidelines or constraints

→ Centered around an artifact presented

→ Add supplementary materials to leave people with something

[24] Image source

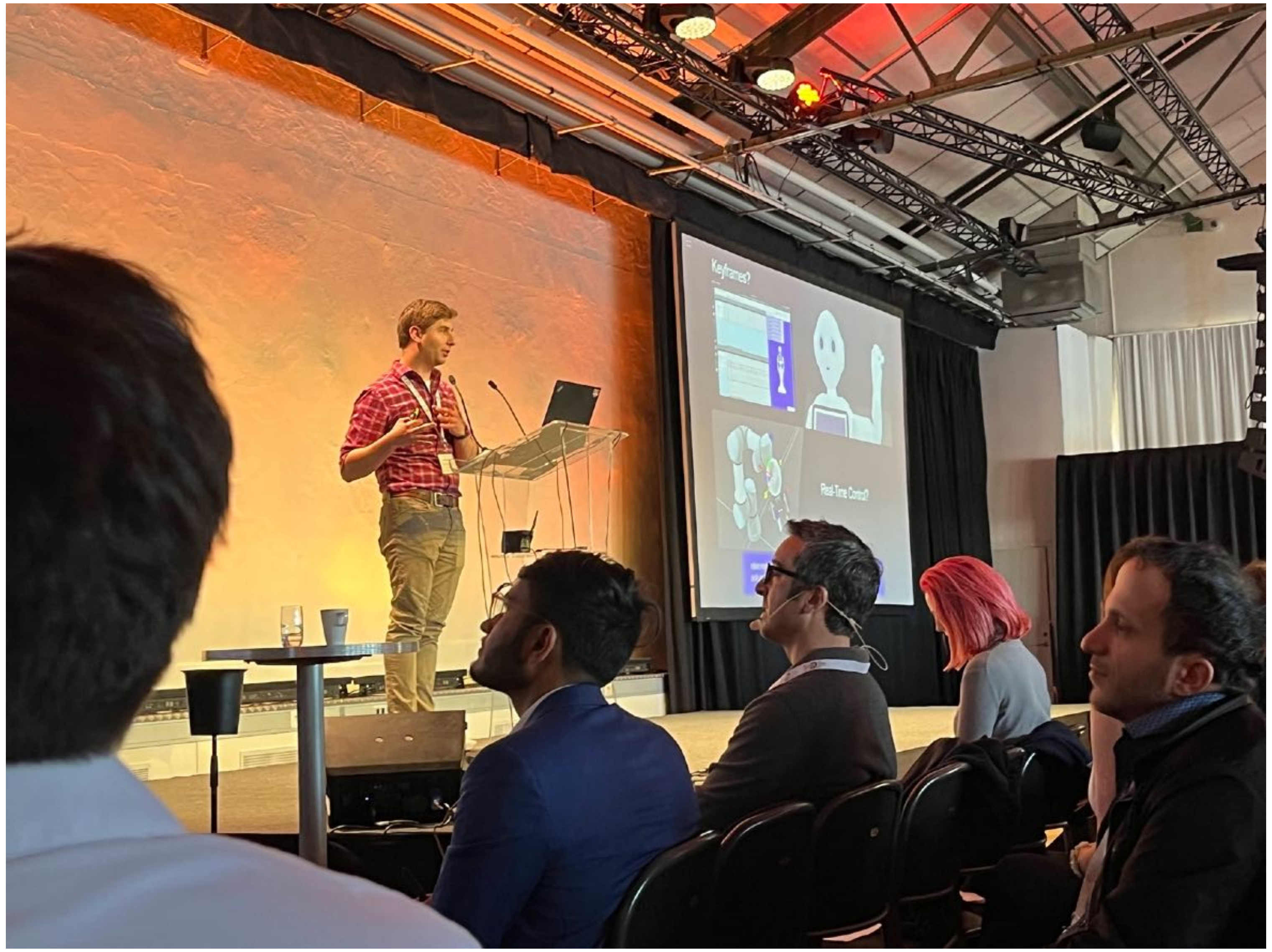
Key challenges & competencies

Oral presentation:

→ Visual storytelling to craft a compelling presentation

→ Public speaking — fluency, body language, handling the unexpected, responding to questions

→ Technical fluency to manage the technology



Poster presentation:

- → Visual design, layout, type choice
- → Printing, transporting, handling a poster
- → Engaging with people during the poster session

Demo presentation:

→ Experience design to present a research artifact

→ Technical competency to setup, operate, and troubleshoot technology

→ Engaging visitors during the demo session

Online Content

- → Types of materials
  - → Videos on social media (e.g., YouTube)
  - → Promotion of work in social media posts (e.g., Twitter)
  - → Websites/blogs that compile materials[25]
- → No specific format or guidelines, but community practices emerge

[25] Do As I Can, Not As I Say

What did we cover today?

- → Course evaluations — Don't forget the treats 🍪🍩
- → Basics of evaluation
- → Forms of evaluation
- → Types of data
- → Types of analyses
- → Basics of communication
- → Forms of communication

Next Steps

- → HACK 5 office hours
  - → My office hours today (2:15-3 pm; CS 6381)
  - → Andy Schoen (2-4 pm; CS 3351)
  - → Hunter Zhang (3-5 pm; CS 3351)
- → HACK 5 demos on Wednesday
- → INTEGRATE final deliverable next Monday