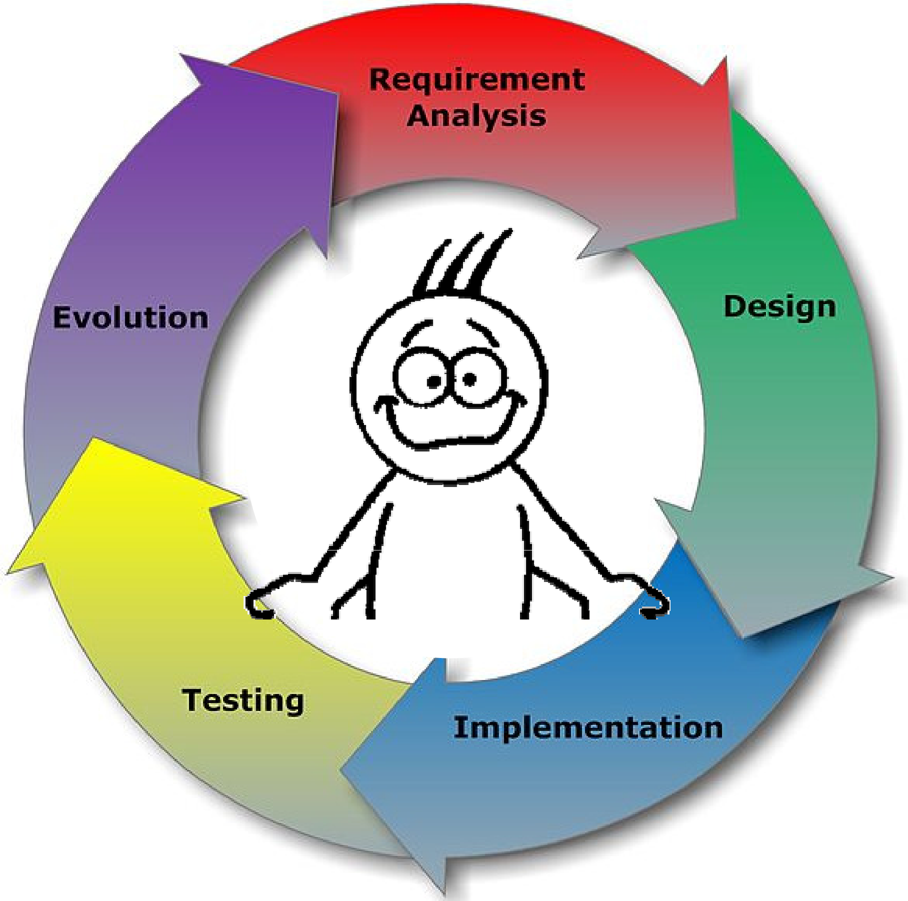



# End Users' Place in the System's Lifecycle

Rebecca Fiebrink
Assistant Professor of Computer Science, also Music
Princeton University

# In what ways might end users engage with “intelligent” interfaces or systems?

## Why? How? What are the consequences?

- Saleema Amershi & James Fogarty: End users training concept classifiers
- Rebecca Fiebrink & Dan Trueman:  End users building interactive music systems
- Dustin Smith & Henry Lieberman: Improving transparency and efficacy of natural language interfaces
- Outline:
  - Related work
  - Paper overviews
  - Broader discussion

# Related topics

End-user Programming

End-user SW Engineering

Programming by Example/Demo.

Interactive Machine Learning

SW Engineering for Intelligent Systems

- Support people in creating systems for themselves

# Related topics

End-user Programming

End-user SW Engineering

Programming by Example/Demo.

Interactive Machine Learning

SW Engineering for Intelligent Systems

- Support people in creating **quality** systems for themselves
- Current research:
  - Understanding how end-user SE differs from professional SE
  - Developing tools to support users without [just] educating them
    - E.g., Whyline, DENIM, Spreadsheet assertions
  - Understanding larger context of end-user development

*Ko et al. "The State of the Art in End-User Software Engineering", 2011*

# Related topics

End-user Programming

End-user SW Engineering

Programming by Example

Interactive Machine Learning

SW Engineering for Intelligent Systems

- A mechanism for end-user programming
- May or may not be intelligent
- Intelligence may or may not be exposed to the user

*Lieberman (ed.): "Your Wish is My Command…" 2001*

# Related topics

End-user Programming

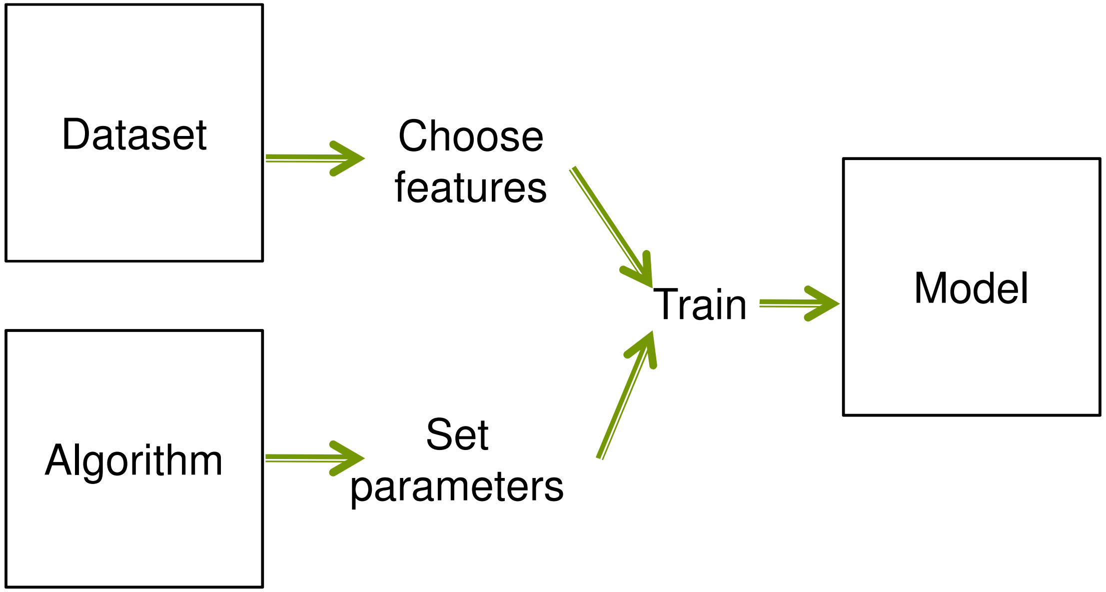
End-user SW Engineering

Programming by Example

Interactive Machine Learning

SW Engineering for Intelligent Systems

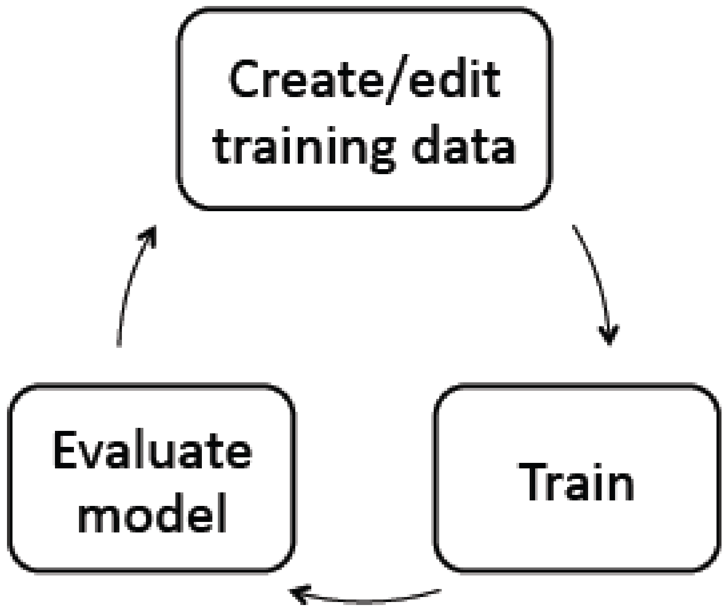
- Inserts user interaction into learning process



# Related topics

- End-user Programming
- End-user SW Engineering
- Programming by Example
- Interactive Machine Learning
- SW Engineering for Intelligent Systems

- Inserts user interaction into learning process



# Related topics

End-user Programming

End-user SW Engineering

Programming by Example

Interactive Machine Learning

SW Engineering for Intelligent Systems

- Inserts user interaction into learning process
- Current research:
  - New interfaces that support more effective control, understanding & feedback
    - E.g., Baker 2009
  - Algorithms that expose different interactive affordances to the user
    - E.g. ManiMatrix, EnsembleMatrix
  - Study of impact of IML as a tool in end-user systems-building and data analysis
    - E.g., Exemplar

# Related topics

End-user Programming

End-user SW Engineering

Programming by Example

Interactive Machine Learning

SW Engineering for Intelligent Systems

- Intelligence introduces new complexities into the software engineering process
- Current research
  - End-user assessment, debugging
    - E.g. Kulesza et al. 2009, Stumpf et al. 2009
  - Supporting machine learning within the larger software development process
    - Kayur Patel et al. 2008–2011

# Shared challenges

- When end user is the system designer, verifier, debugger, etc. of an intelligent system:
  - How transparent should the system be to the user?
  - How can system state, available actions, likely outcomes be made more transparent?
  - What are the means by which the user can exercise control or influence?
  - How can the system support good software engineering practice without distracting the user from his or her “real” goals?

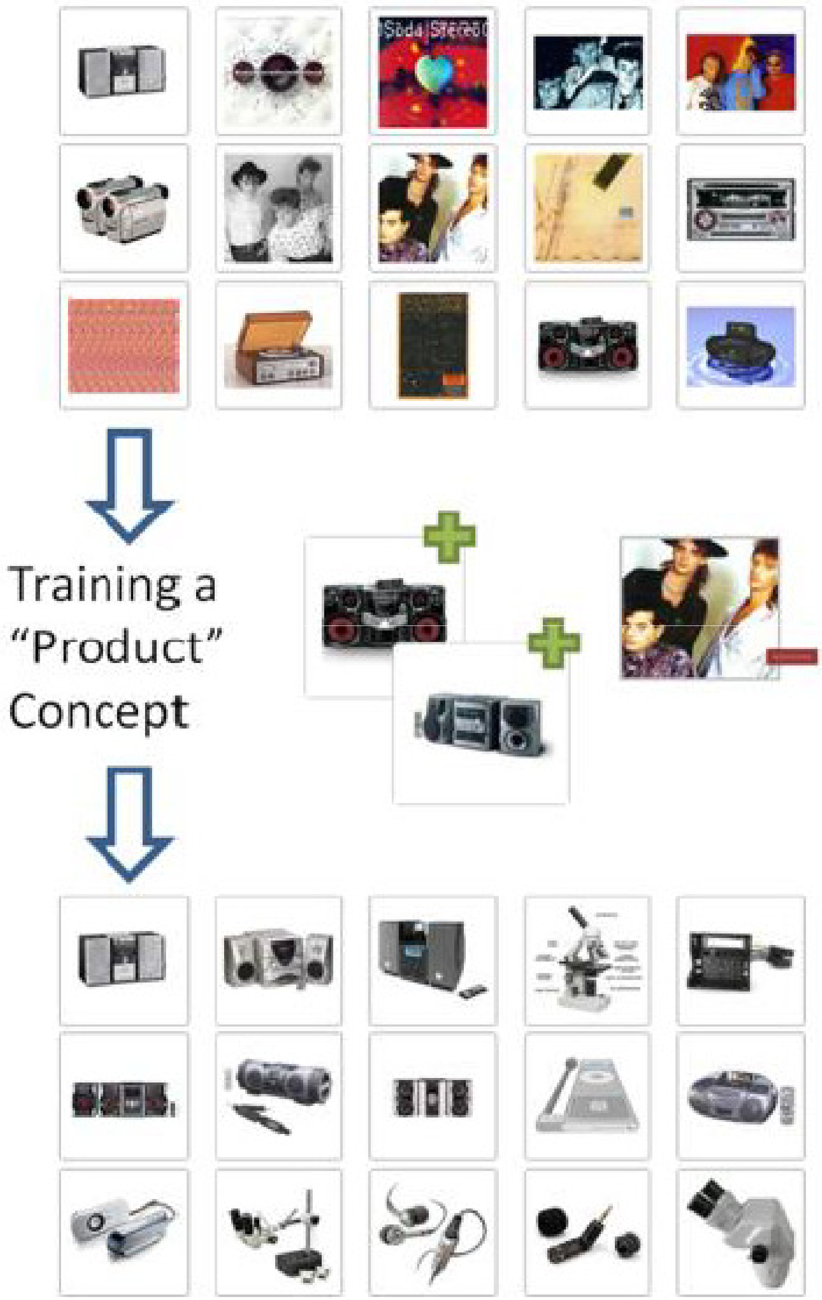
# Amershi & Fogarty

"… Just as the past twenty years have seen the emergence of search, we expect the next twenty years will see machine learning embedded in every aspect of end-user interaction."

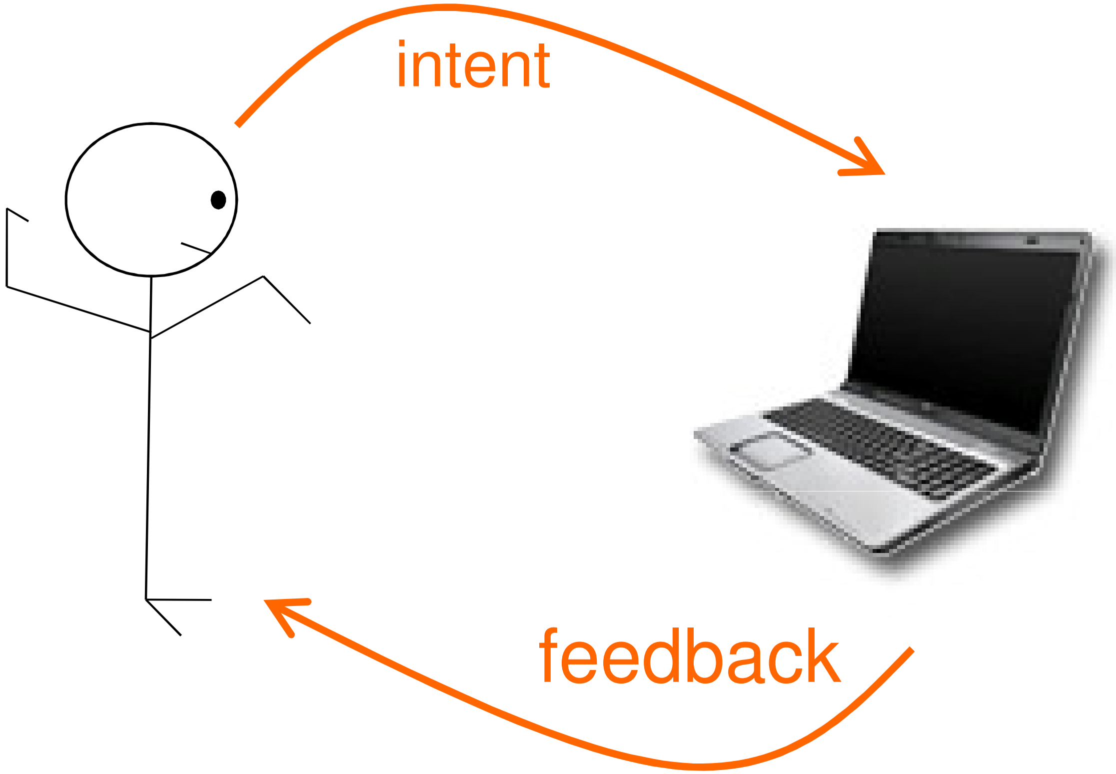
- 3 systems
  - CueFlik
  - CueT
  - ReGroup

# CueFlik

Fogarty, Tan, Kapoor,
and Winder
CHI’08



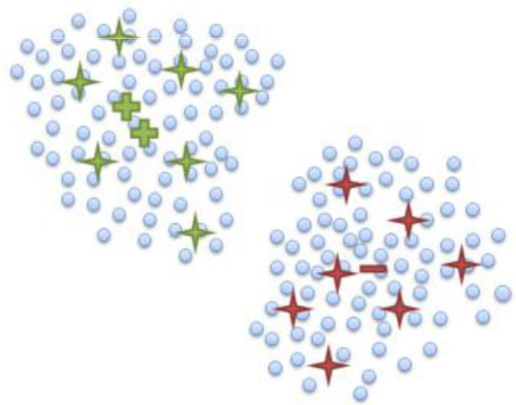
# Interactive machine learning in CueFlik




- Intent and feedback both presented through examples
- Similar approaches:
  - Crayons (Fails & Olson 2003)
  - CueTip (Shilman et al. 2006)
  - Wekinator (Fiebrink 2008+)

# Findings:
# Supporting effective control & feedback

- CHI ‘08: Showing users the high-certainty **positive and negative examples** provide better feedback than single view ranked by positive likelihood
- UIST ‘09: Providing users with **overviews** of positive and negative regions leads to better training example selection

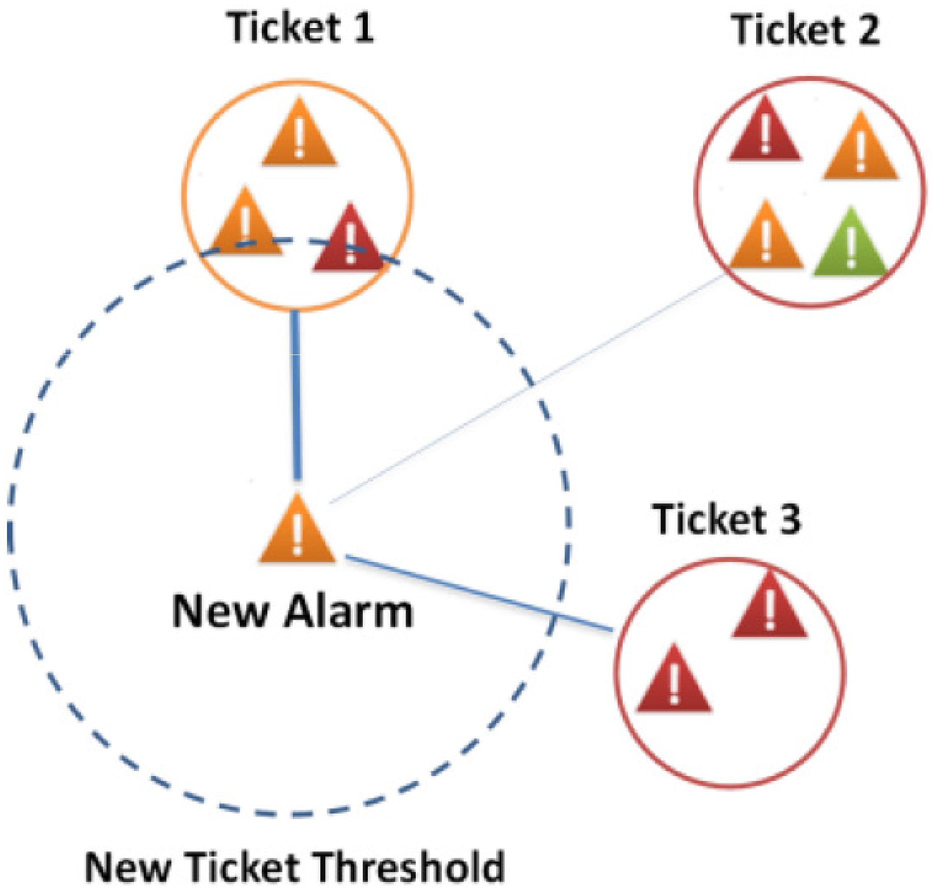


- CHI ‘10: Allowing more flexible exploration of **alternative** training example sets– and their consequences– improves user comfort and model quality

# CueT

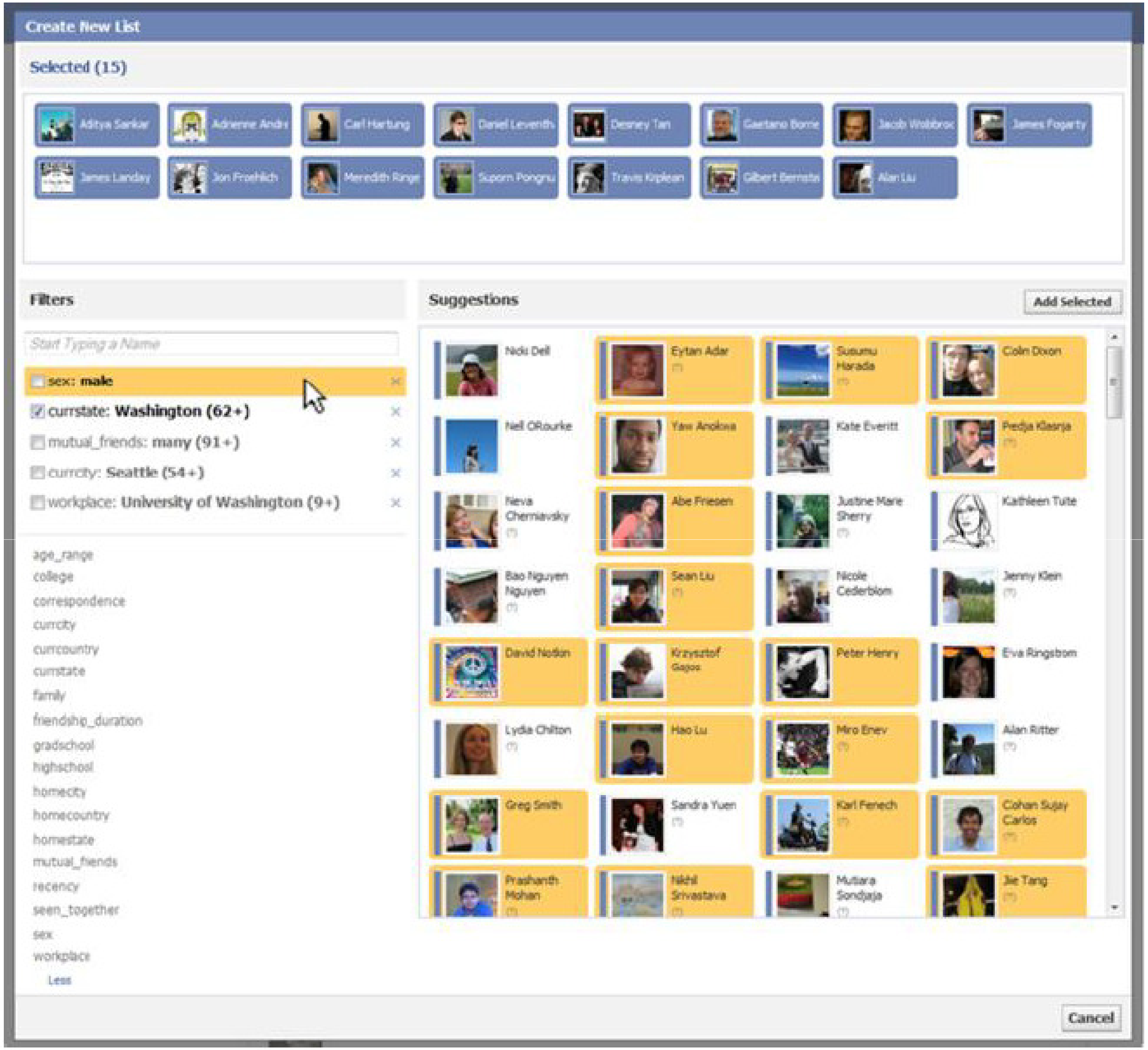
Amershi, Lee, Kapoor, Mahajan, Christian
CHI '11

- Interactive clustering of examples in a stream
  - Example = A network alarm
  - Cluster = A set of related alarms, grouped into one ticket
- **Distance metric and threshold are learned** from user actions
- **Visual feedback**



# ReGroup

Amershi, Fogarty, and Weld
CHI '12



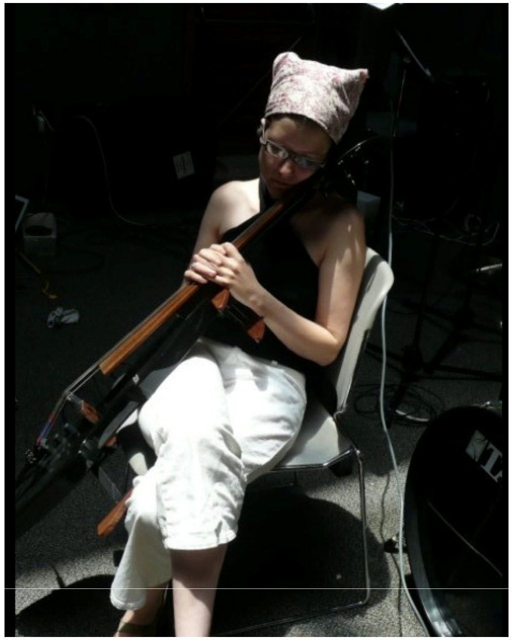
# Summary: Amershi & Fogarty

- Machine learning supports end-user **interactions with data**
- End-users can train **composable concepts** as part of configuring more sophisticated data processing and manipulation

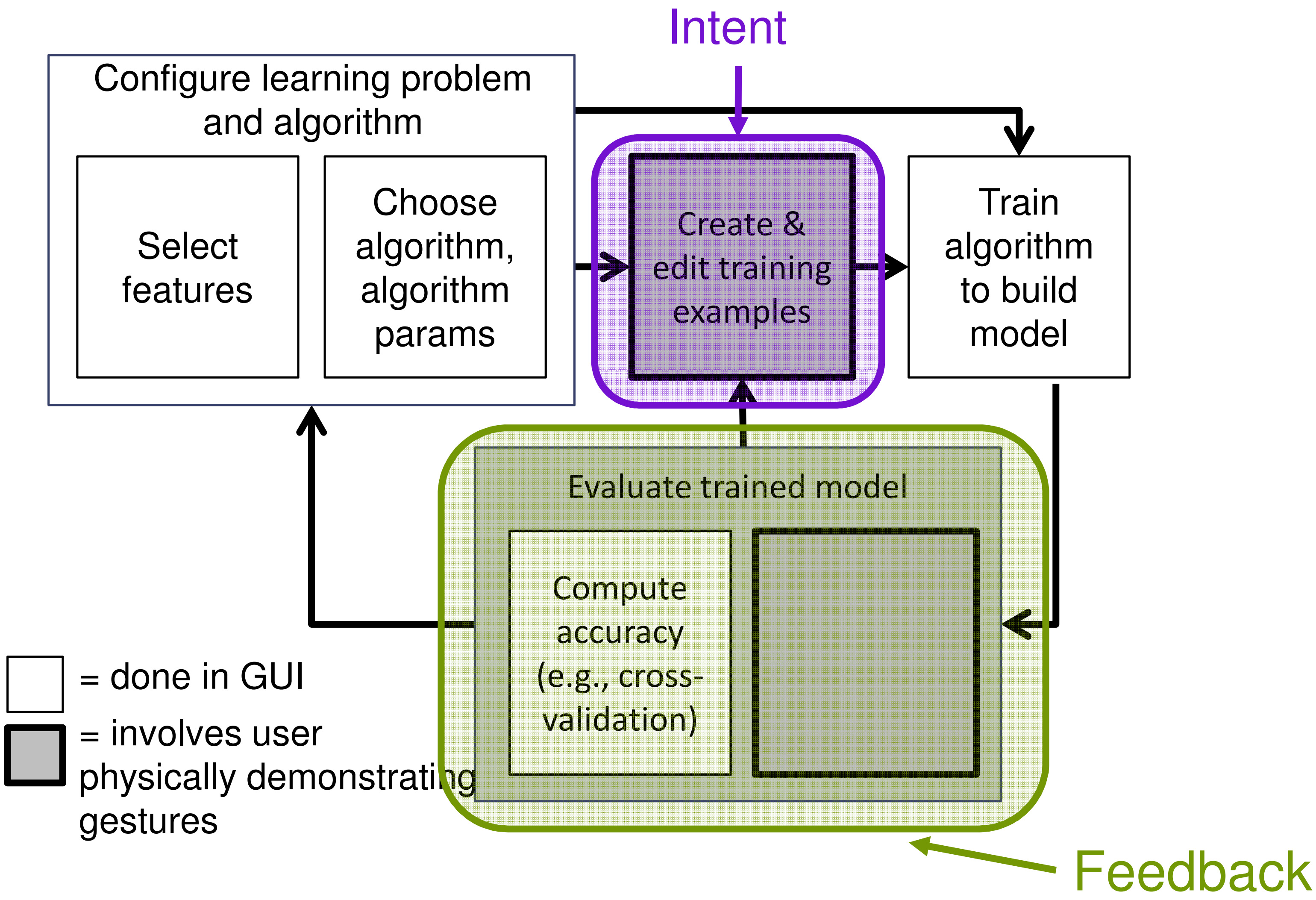
# Fiebrink and Trueman

- Goal: Enable **musical end users** to apply machine learning to their work
  - E.g., real-time gestural recognition and control systems

- Outline:
  - Overview of the Wekinator
  - Five key findings of interest to the workshop



# The Wekinator workflow




Fiebrink et al., 2009+

# Observations

- 1. **Users have diverse goals for models**, extending far beyond model accuracy
  - Generalizing from training data may not be [most] important
- 2. **Interactive machine learning can "train" the user** in unexpected— and possibly advantageous—ways.
  - Users can learn about what is accurately / quickly learnable by the algorithm given the features
- 3. **Users can employ knowledge about what is learnable to adjust the learning concept** they choose to teach the system

Fiebrink, Cook, and Trueman, CHI '11

## More observations: Creativity & Embodiment

- 1. **Interactive machine learning can support creative work**
  - Rapid prototyping
  - Exploring alternatives
  - Sketching

- **2. Interactive machine learning can support an embodied approach to design**
  - "feel" can be encoded in training data and can be part of evaluation
  - Design in the physical world, without requiring math/code

# Dustin Smith & Henry Lieberman

- "… when a person speaks to a computer, … it is unclear what the computer knows and how the computer will use its knowledge to interpret [his] utterance. This lack of transparency and predictability are barriers for NLIs."
- "… augmenting a NLI with non-linguistic modalities may present a mode for recovering from interpretation failures that is richer than relying solely on dialogue."

  - Inevitability of failure
  - Information required to deal with failure
  - Research questions
  - Research system: NLI calendering interface for event descriptions

# Failure is inevitable


Divergent beliefs
Divergent goals

(Not a human.)

# What information does the user require?

**What** the interface knows
- words
- word senses
- entities it can refer to (context set)

**How** the interface derived an interpretation
- Senses and syntactic structures
- Pragmatic assumptions
- Subset of structures that were referenced

- **NLI designers** should make these explicit
- **Users** can employ this information to improve the interpreter

## Research Questions

- When should the system request clarification?
- How does the interpreter’s transparency affect the interface’s overall usability?
  - How does it affect the work the user perceives the interface to be doing?
  - Does knowing the status of the overall interpretation help the user to achieve the communication goal?
  - Does having control over the interpretation decisions help the user achieve the communication goal?

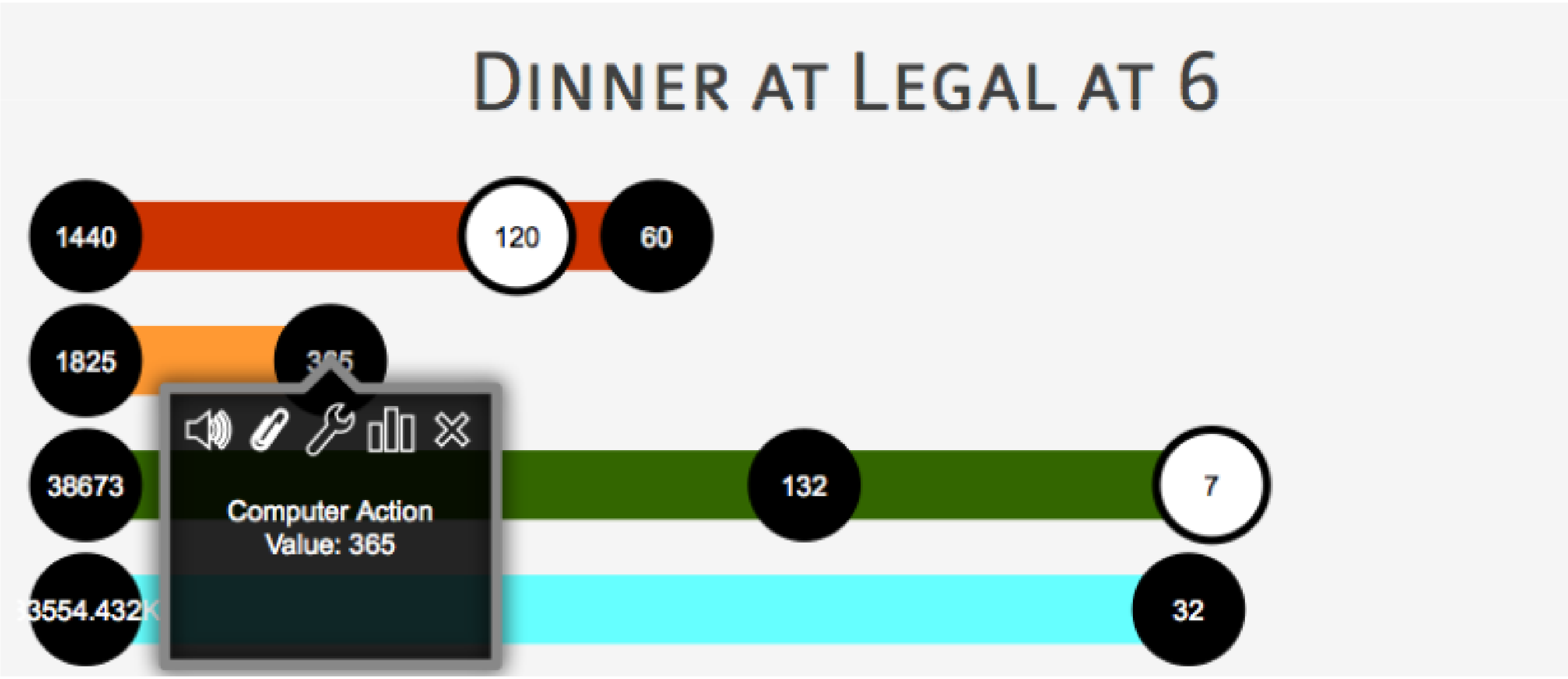
# The event description system

- Given:
  - User's written description of an event
  - System knowledge and assumptions
  - User interaction
- Goal is precise agreement on:
  - Starting time
  - Day
  - Location
  - Group of people

"Dinner with Bill's friends at 6 nearby"

# Problem representation & interaction

- At any point, system knows of one or more allowable events.
- A sequence of system assumptions and/or user actions narrows the set down to one.



# Summary: Smith & Lieberman

- Introduce more transparency and room for error correction and guidance into NLIs
  - Starting from an understanding of the information a user needs and how an interpreter could be designed to supply that information
- Not just end-user debugging, understanding, and configuration
  - Longterm goal: collect culture-specific lexical & semantic knowledge and use it to improve NLIs in the future

# Discussion points: End users in the lifecycle

- Examples of **why** end users might engage in creating, improving, debugging intelligent software
  - Concept classifiers for managing data
  - Building interactive systems
  - Improving NLIs they want to use
- Demonstrations of **how** end users can be engaged in creating and improving intelligent software
  - Labeling examples, creating examples, evaluating models
  - Correcting and improving machine assumptions
- **Positive consequences** of this engagement
  - More accurate and effective systems
  - More efficient, satisfying system-building process
  - Better knowledge of the learning system
- **Research challenges**:
  - Not burdening the user with extra detail & responsibility
  - Supporting debugging of intelligent systems
  - Eliciting and supporting a range of user goals
  - Identifying universal principles & approaches across application domains