Learning Analytics

Dr. Bowen Hui

Computer Science

University of British Columbia Okanagan

Data About Users

- What kind of data can we collect?

Data About Users

- What kind of data can we collect?
  - Numbers, images, words, …
  - Computer events (e.g., keystrokes, clicks)
  - Abstracted behaviours (e.g., words/topics, distracted actions, mistakes made)

Purpose

- What is data used for?
  - Providing information on people/things of interest
  - Finding correlations and patterns
  - Informing decision making

- What data should we collect?
  - Depends on the problem at hand!

Exercise: Website Preferences

- What data can you collect to figure out what kinds of websites your user likes?
  Be as specific as possible.

Web Data

- Categories of web data:
  - Server logs
  - Content data
  - Website structure
  - Demographics / Historical

Web Data

- Server logs
  - user’s IP address
  - user’s authentication name
  - data-time stamp of access
  - HTTP request
  - response status
  - size of requested resource
  - referrer URL (the page the user came from)
  - user’s browser identification

Web Data

- Content data
  - includes all the page content accessible by user:
  - text
  - images
  - any other multimedia content

Web Data

- Website structure (why?)
  - available sitemap of a website
  - dual roles:
    learn structure info or generate structure

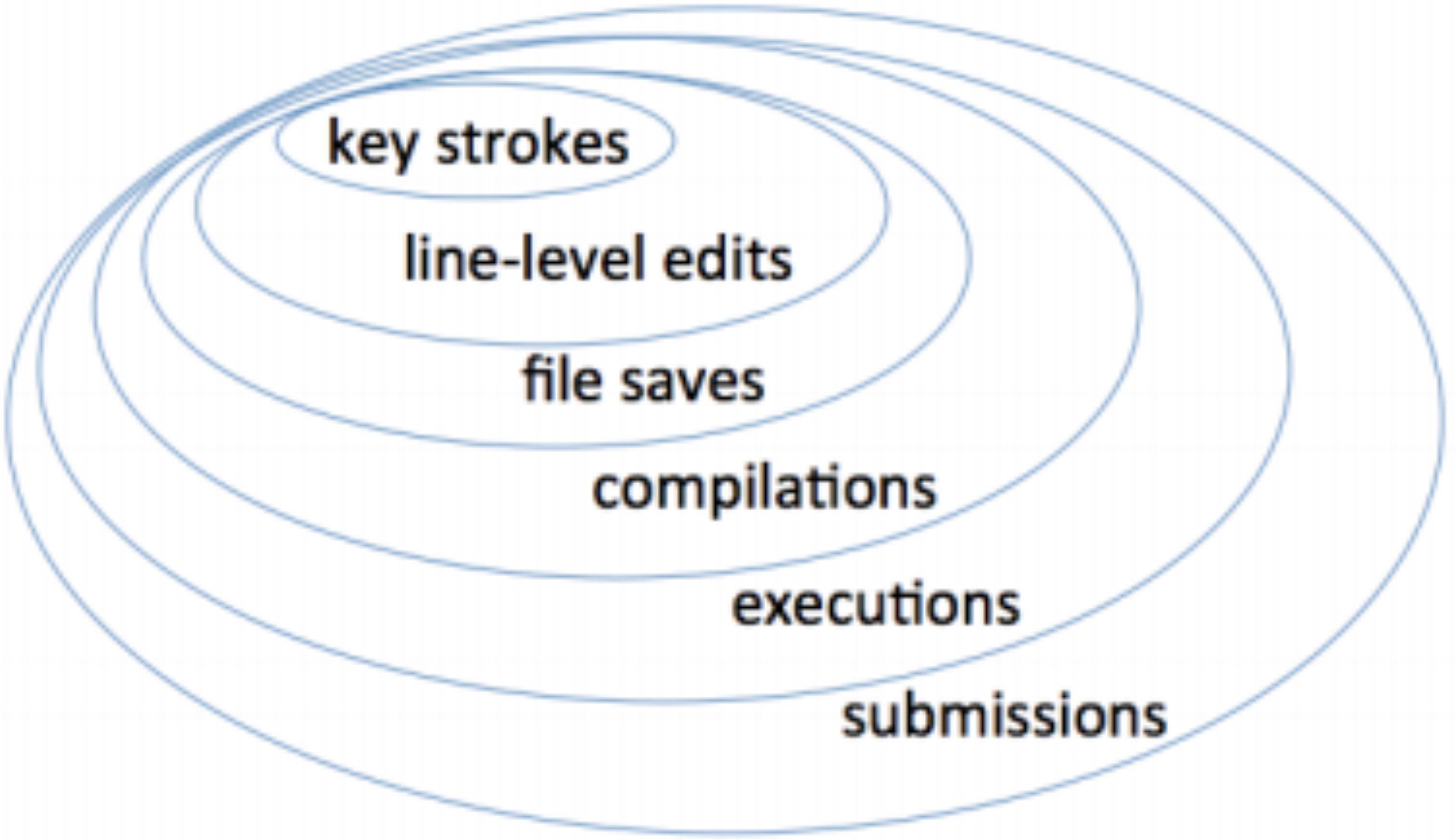
Web Data

- Demographics / Historical
  - explicitly given by users
  - today: data from third-party apps
    - Social media integration
    - What people like (e.g. purchases)
    - What people do (other sites/apps they use)

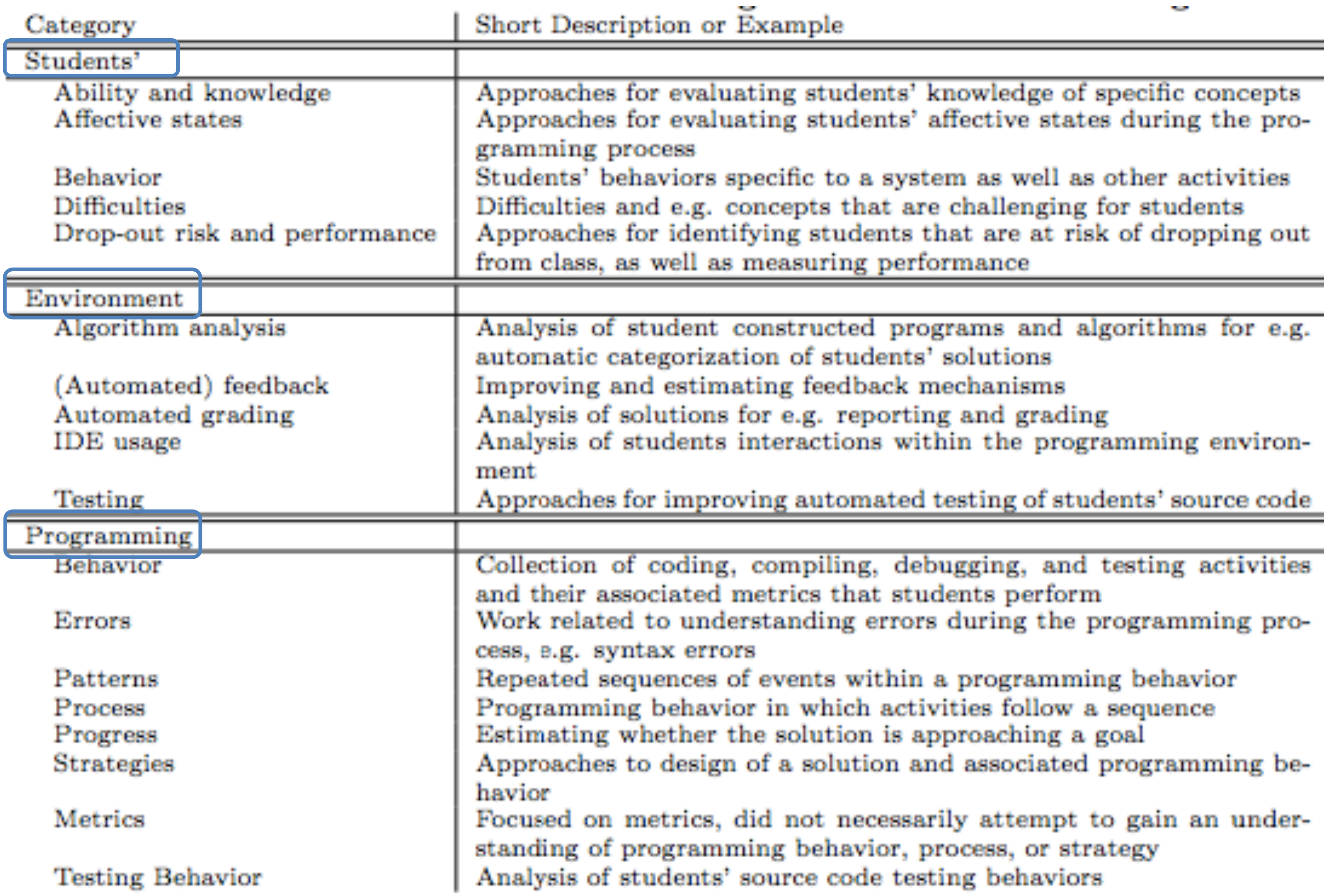
Exercise: Programming Difficulty

- What data can you collect to figure out which concepts/steps students are having a hard time with in a programming assignment? Be as specific as possible.

Coding Data



Reference: Educational Data Mining and Learning Analytics in Programming: Literature Review and Case Studies (Ihantola et al. 2015)



Category	Short Description or Example
Students'	
Ability and knowledge	Approaches for evaluating students' knowledge of specific concepts
Affective states	Approaches for evaluating students' affective states during the programming process
Behavior	Students' behaviors specific to a system as well as other activities
Difficulties	Difficulties and e.g. concepts that are challenging for students
Drop-out risk and performance	Approaches for identifying students that are at risk of dropping out from class, as well as measuring performance
Environment	
Algorithm analysis	Analysis of student constructed programs and algorithms for e.g. automatic categorization of students' solutions
(Automated) feedback	Improving and estimating feedback mechanisms
Automated grading	Analysis of solutions for e.g. reporting and grading
IDE usage	Analysis of students interactions within the programming environment
Testing	Approaches for improving automated testing of students' source code
Programming	
Behavior	Collection of coding, compiling, debugging, and testing activities and their associated metrics that students perform
Errors	Work related to understanding errors during the programming process, e.g. syntax errors
Patterns	Repeated sequences of events within a programming behavior
Process	Programming behavior in which activities follow a sequence
Progress	Estimating whether the solution is approaching a goal
Strategies	Approaches to design of a solution and associated programming behavior
Metrics	Focused on metrics, did not necessarily attempt to gain an understanding of programming behavior, process, or strategy
Testing Behavior	Analysis of students' source code testing behaviors

Reference: Educational Data Mining and Learning Analytics in Programming: Literature Review and Case Studies (Ihantola et al. 2015)

Types of Data

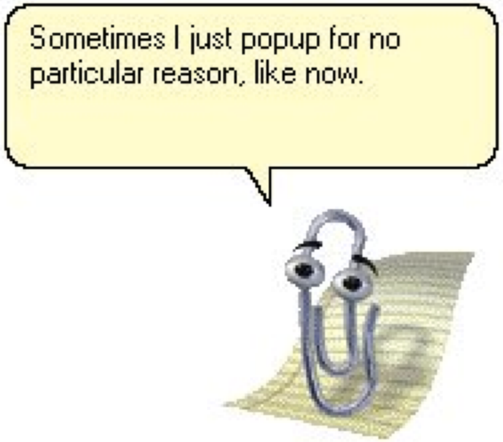
- The kind of data we collect depends heavily on the problem of interest
- Implicit feedback
  - E.g.: clicked links, content of web pages read, errors made, button presses
- Explicit feedback
  - E.g.: labeling interesting web pages, rating movies, completing survey

Key Considerations

- Relevance of the data
  - Having as much data as possible isn’t necessarily the best approach
- Granularity
  - Data should be defined at consistent levels (e.g. compare “browsing behaviour” vs. a key press)
- Age
  - Is data outdated? Should newer data be weighted more heavily than older data?

Aggregating Data

- How to aggregate data to define abstracted behaviours?
- Case study: Lumiere/Excel user study (1998)



Sometimes I just popup for no particular reason, like now.

How did the authors use the Wizard of Oz study to collect data?

Aggregating Data

- How to aggregate data to define abstracted behaviours?
- Case study: Lumiere/Excel user study (1998)
- Gauge the ability experts can guess user’s goal by simply watching the user’s actions through a “keyhole” interface
- Interested in seeing how experts infer:
  - Likelihood a user needs help
  - The type of help the user needs

Inferring User Needs: Classes of Evidence

- Search
- Focus of attention
- Introspection
- Undesired effects
- Inefficient command sequences
- Domain-specific syntactic and semantic content

Reference: The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users (Horvitz et al. 1998)

Inferring User Needs: Classes of Evidence

- Search
  - Repetitive, scanning patterns associated with attempts to search for or access an item or function
  - E.g.: user exploring multiple menus, scrolling through text, mousing over and clicking on multiple non-active regions
- Focus of attention
- Introspection
- Undesired effects
- Inefficient command sequences
- Domain-specific syntactic and semantic content

Inferring User Needs: Classes of Evidence

- Search
- Focus of attention
  - Selection and/or dwelling on graphical objects
  - Dwelling on portions of document or specific subtext after scrolling through document
- Introspection
- Undesired effects
- Inefficient command sequences
- Domain-specific syntactic and semantic content

Inferring User Needs: Classes of Evidence

- Search
- Focus of attention
- Introspection
  - Sudden pause after a period of activity
  - Significant slowing of rate of interaction
- Undesired effects
- Inefficient command sequences
- Domain-specific syntactic and semantic content

Inferring User Needs: Classes of Evidence

- Search
- Focus of attention
- Introspection
- Undesired effects
  - Attempts to return to a prior state after an action
  - E.g.: Undoing recent action (execute undo command, closing dialog box shortly after it is opened)
- Inefficient command sequences
- Domain-specific syntactic and semantic content

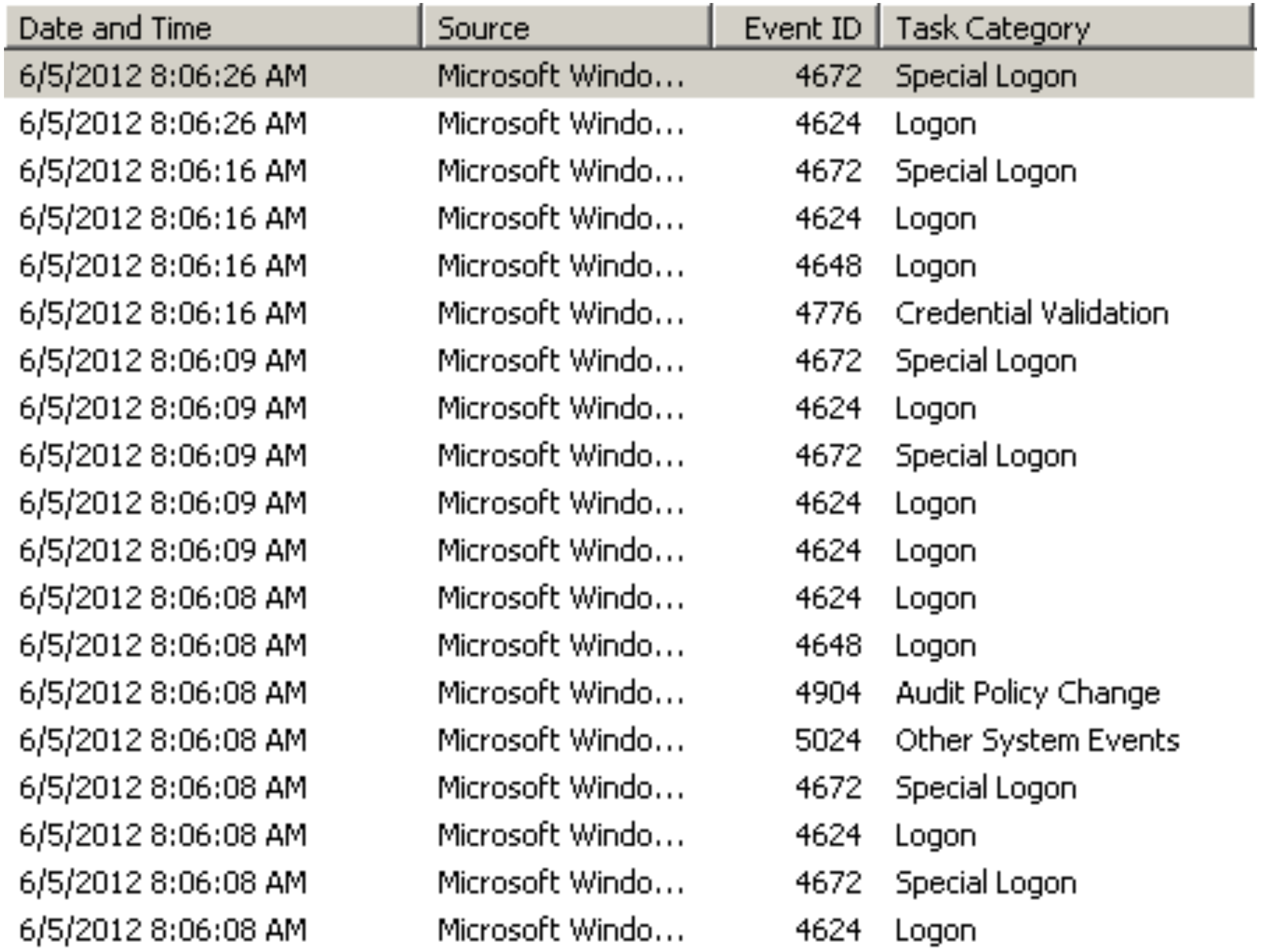
Inferring User Needs: Classes of Evidence

- Search
- Focus of attention
- Introspection
- Undesired effects
- Inefficient command sequences
  - User performing operations that could be done more simply or efficiently via alternate sequence
- Domain-specific syntactic and semantic content

Inferring User Needs: Classes of Evidence

- Search
- Focus of attention
- Introspection
- Undesired effects
- Inefficient command sequences
- Domain-specific syntactic and semantic content
  - Consideration of special distinctions in content or structure of documents and how user interacts with these features

Bridging the Gulf between System Events and User Actions

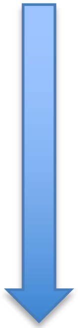


Date and Time	Source	Event ID	Task Category
6/5/2012 8:06:26 AM	Microsoft Windo...	4672	Special Logon
6/5/2012 8:06:26 AM	Microsoft Windo...	4624	Logon
6/5/2012 8:06:16 AM	Microsoft Windo...	4672	Special Logon
6/5/2012 8:06:16 AM	Microsoft Windo...	4624	Logon
6/5/2012 8:06:16 AM	Microsoft Windo...	4648	Logon
6/5/2012 8:06:16 AM	Microsoft Windo...	4776	Credential Validation
6/5/2012 8:06:09 AM	Microsoft Windo...	4672	Special Logon
6/5/2012 8:06:09 AM	Microsoft Windo...	4624	Logon
6/5/2012 8:06:09 AM	Microsoft Windo...	4672	Special Logon
6/5/2012 8:06:09 AM	Microsoft Windo...	4624	Logon
6/5/2012 8:06:09 AM	Microsoft Windo...	4624	Logon
6/5/2012 8:06:08 AM	Microsoft Windo...	4624	Logon
6/5/2012 8:06:08 AM	Microsoft Windo...	4648	Logon
6/5/2012 8:06:08 AM	Microsoft Windo...	4904	Audit Policy Change
6/5/2012 8:06:08 AM	Microsoft Windo...	5024	Other System Events
6/5/2012 8:06:08 AM	Microsoft Windo...	4672	Special Logon
6/5/2012 8:06:08 AM	Microsoft Windo...	4624	Logon
6/5/2012 8:06:08 AM	Microsoft Windo...	4672	Special Logon
6/5/2012 8:06:08 AM	Microsoft Windo...	4624	Logon

Image taken from scottlinux.com

General Idea

- Time stamp to indicate recency
- Low-level atomic events
(click on “File”, closed Font dialog box, …)

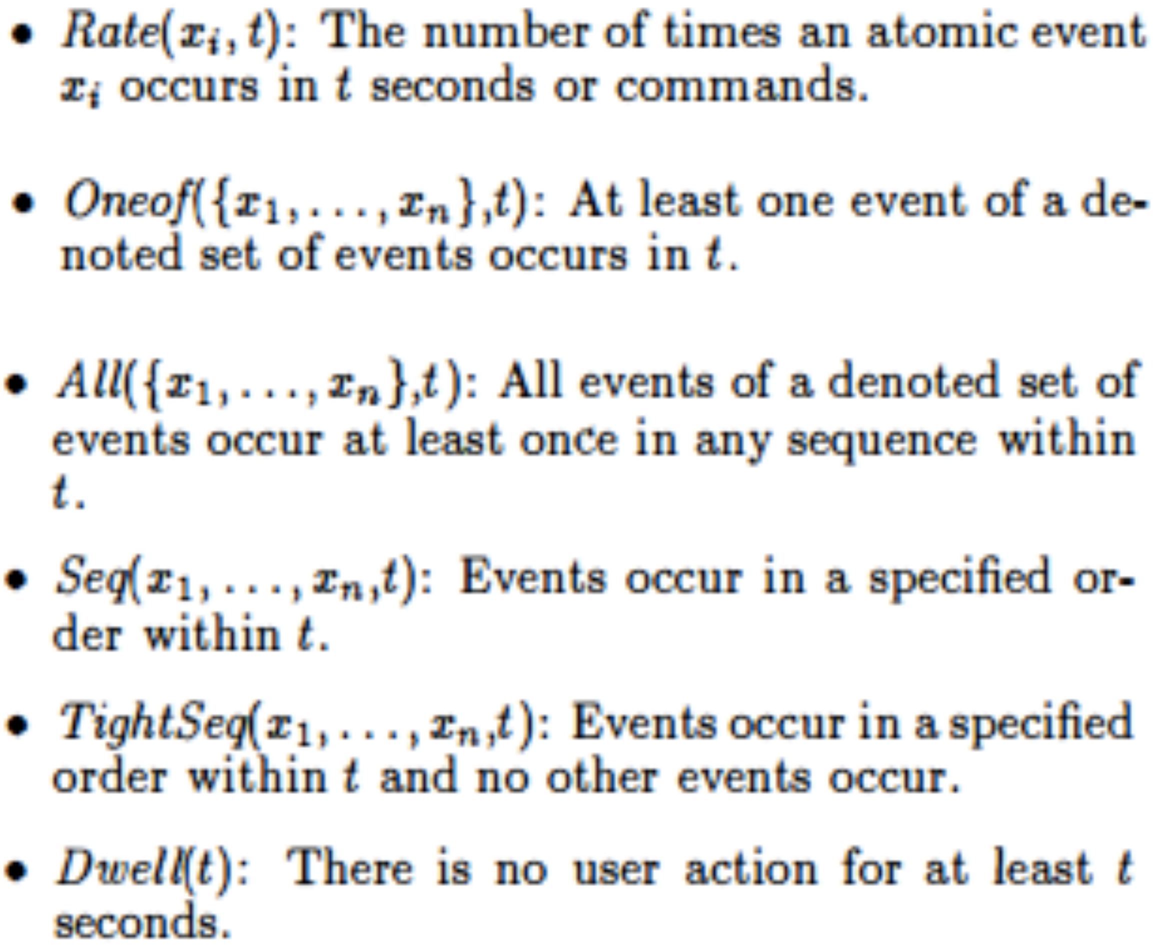


Automatic translation

- Higher-level semantic user actions
(menu surfing, mouse meandering, …)

Lumiere Events Language

- Primitives provided in the language:



  - *Rate*(x_i, t): The number of times an atomic event x_i occurs in t seconds or commands.
  - *Oneof*($\{x_1, \ldots, x_n\}$,t): At least one event of a denoted set of events occurs in t.
  - *All*($\{x_1, \ldots, x_n\}$,t): All events of a denoted set of events occur at least once in any sequence within t.
  - *Seq*($x_1, \ldots, x_n$,t): Events occur in a specified order within t.
  - *TightSeq*($x_1, \ldots, x_n$,t): Events occur in a specified order within t and no other events occur.
  - *Dwell*(t): There is no user action for at least t seconds.

Lumiere Events Language

- Models user actions as atomic events directly
  - E.g., how fast the user moved the mouse
- Streams of atomic events can be formed into Boolean and set-theoretic combinations of low-level events
  - E.g., saved a file via toolbar or keyboard shortcut
- Compose new modeled user actions from previously defined events
  - E.g., user dwelled for at least t seconds after a scroll

Alternative Data-Driven Approach

- Gather large amounts of data logs
- (Ideally) have corresponding labeled user actions
- Example of labeled data:

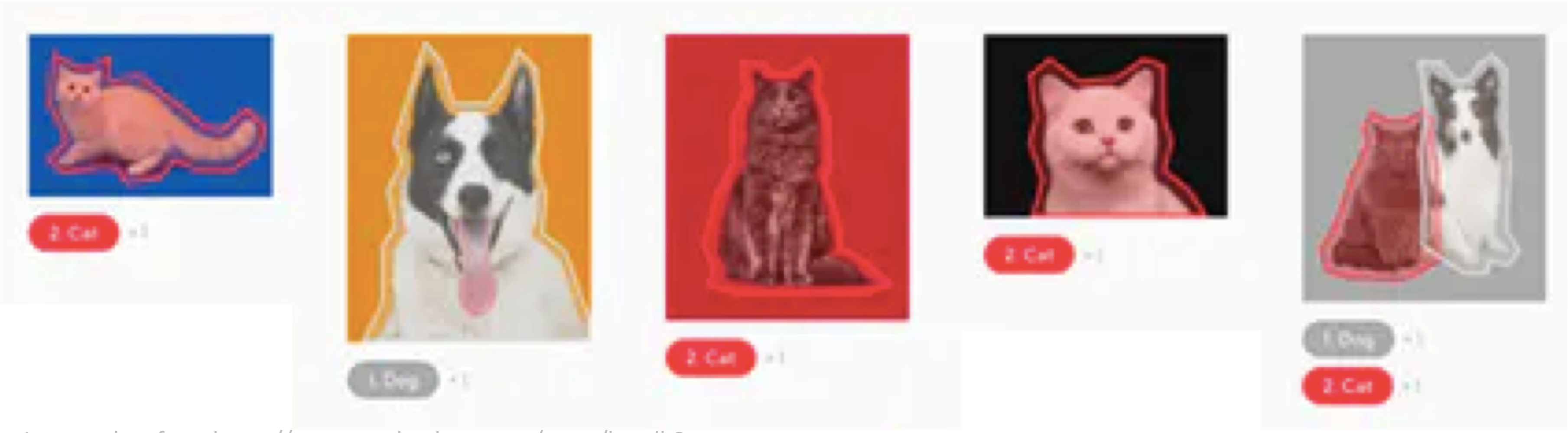
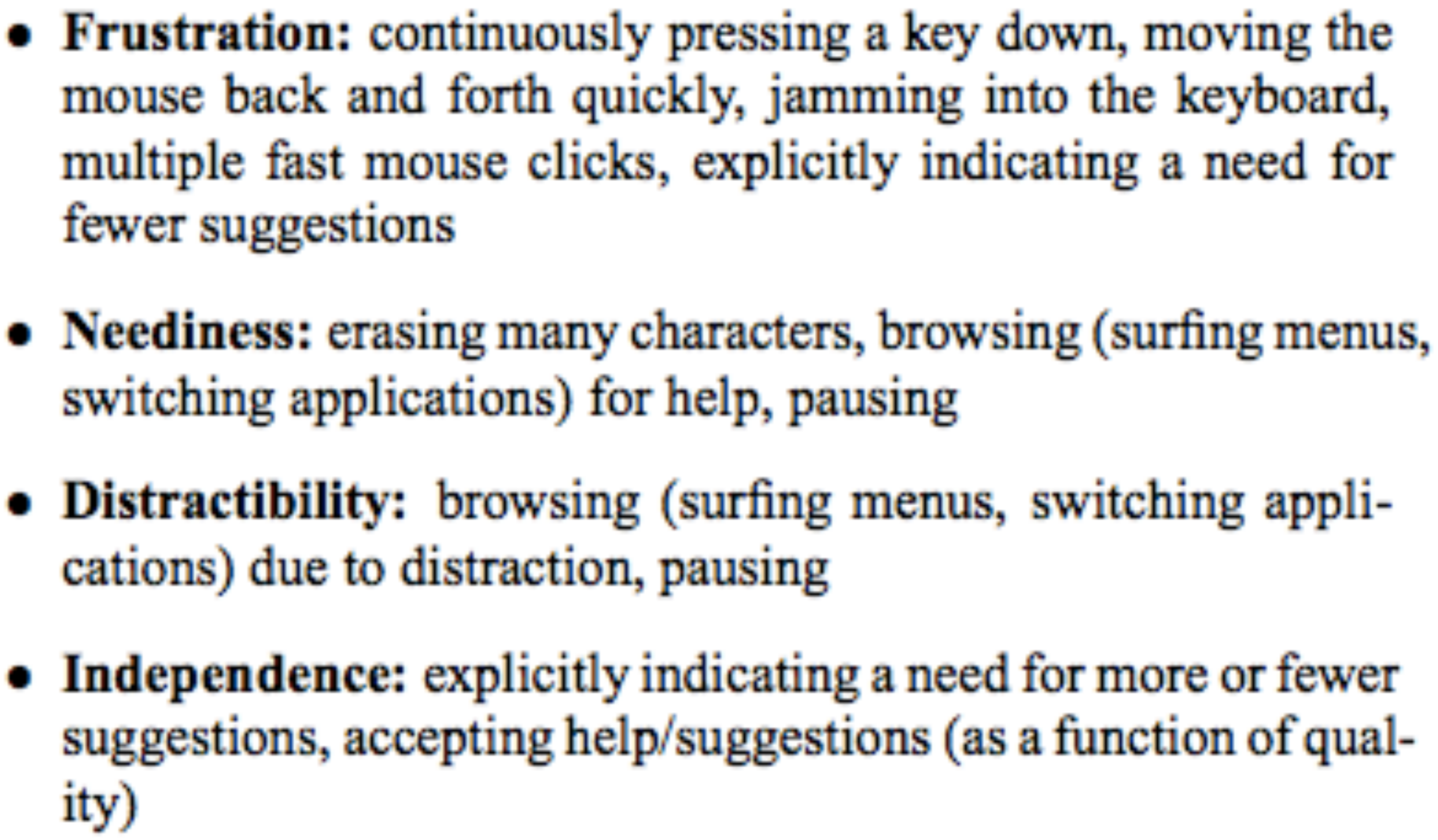


Image taken from https://www.producthunt.com/posts/handl-3

Alternative Data-Driven Approach

- Gather large amounts of data logs
- (Ideally) have corresponding labeled user actions
- Develop patterns of events that map to those user actions
- Analyze accuracy and coverage
- Iterate process to find best set of patterns

Learning User Behaviours



- **Frustration:** continuously pressing a key down, moving the mouse back and forth quickly, jamming into the keyboard, multiple fast mouse clicks, explicitly indicating a need for fewer suggestions
- **Neediness:** erasing many characters, browsing (surfing menus, switching applications) for help, pausing
- **Distractibility:** browsing (surfing menus, switching applications) due to distraction, pausing
- **Independence:** explicitly indicating a need for more or fewer suggestions, accepting help/suggestions (as a function of quality)

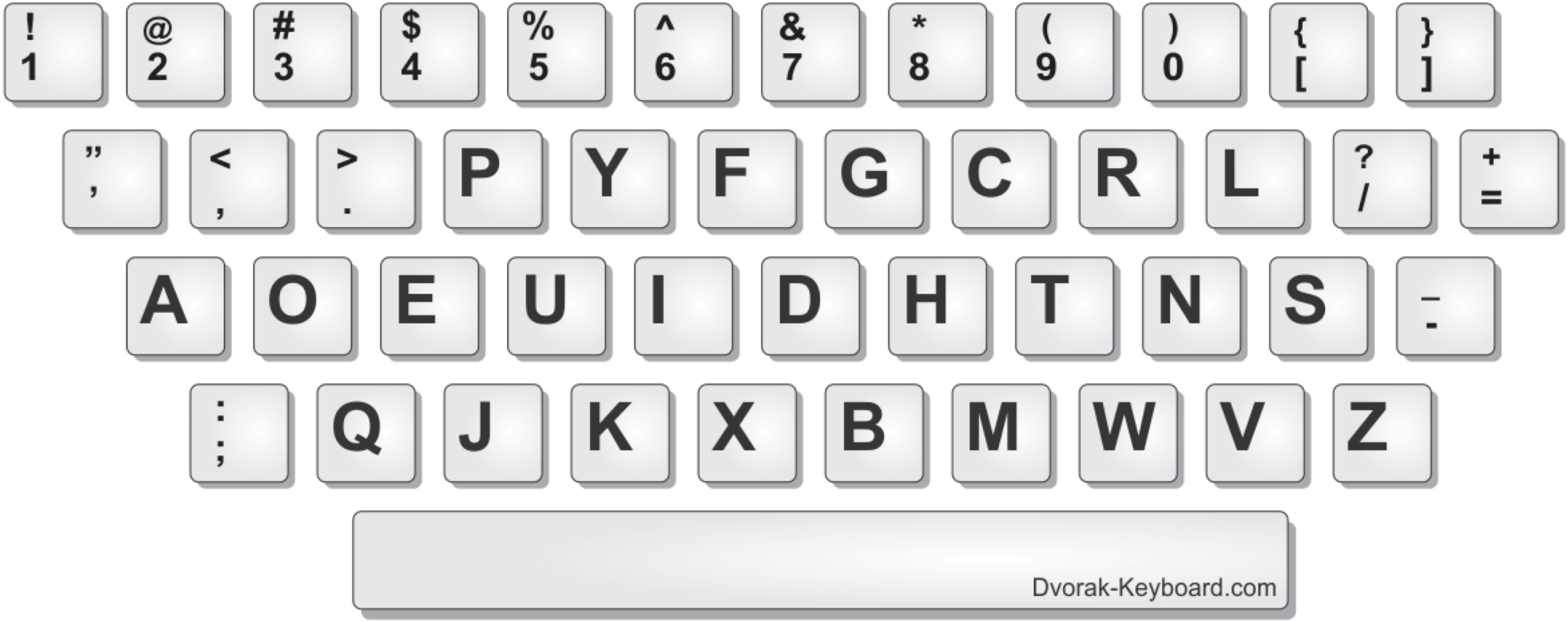
Reference: Who’s Asking for Help? A Bayesian Approach to Intelligent Assistance (Hui & Boutilier 2006)

How to Get Labeled Data?

Image taken from www.clipartkid.com

User Study!

- Typing with dvorak keyboard



Dvorak-Keyboard.com

Experiment Design

How did the authors get labeled data indicating when a person is frustrated, needy, distracted, or independent while the person is typing?

User Study!

- Typing with dvorak keyboard
    - Frustration condition – sticky keys
    - Distraction condition – popups and sounds
    - Neediness condition – difficulty of words
    - Independence – questionnaire and word completion
- Specific intervals, asked user to label current mental state
- Obtained partially labeled data

Data Analysis Step

- Using the collected partially labeled data
  - Extract patterns associated to labeled data for various user conditions
  - Extract similar patterns elsewhere
  - Estimate missing labels

- Now: Completely labeled data
  - Associate extracted patterns to various user conditions probabilistically

Summary

- Different types of data to collect
- Implicit vs. explicit feedback
- Given a problem, identify relevant data to address it
- Mining system events to obtain user behavioural data