Mobile Reference Environment

Rapid Prototyping for Computer Systems

Spring 2006

Carnegie Mellon University

Introduction

Instructors:

Dan Siewiorek
Asim Smailagic

Sponsors:

Dick Martin
Inmedius Software, Inc.

Introduction

HCI Group

Brian Ellis
Anand Gopalkrishnan
Yong Woo Rhee
Aashni Shah
Wen Shu Tang
Dan Zinzow

Project Definition

- This project does *not* deal directly with aircraft maintenance
- Currently deals with the Aircraft Maintenance Environment software (AME)
  - Provides users of AME with a better, more convenient way of learning the system, and retaining and improving upon their knowledge of it

Project Scope

- The important part is the "meta-problem"
  - Training manuals and reference manuals differ in approach, detail, philosophy
- If we can enhance the accessibility of the AME documentation, we can use these techniques on any set of training and/or reference manuals

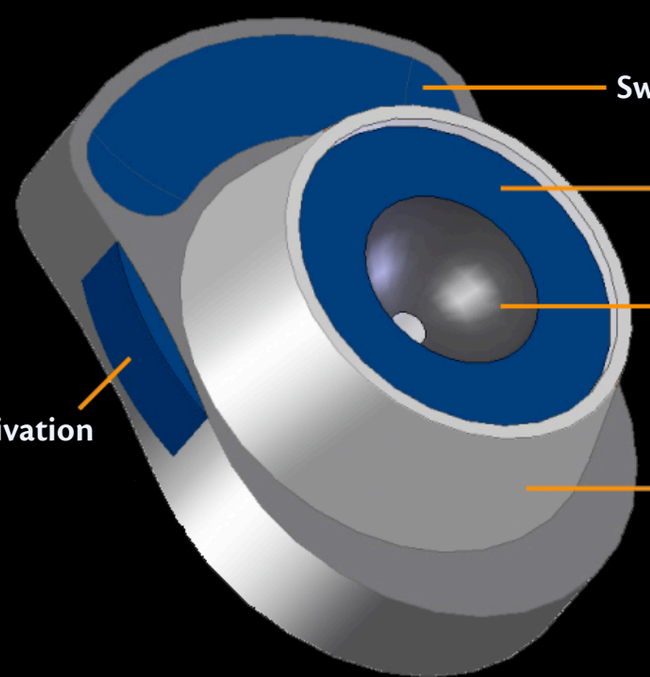
Functional Requirements: Hardware

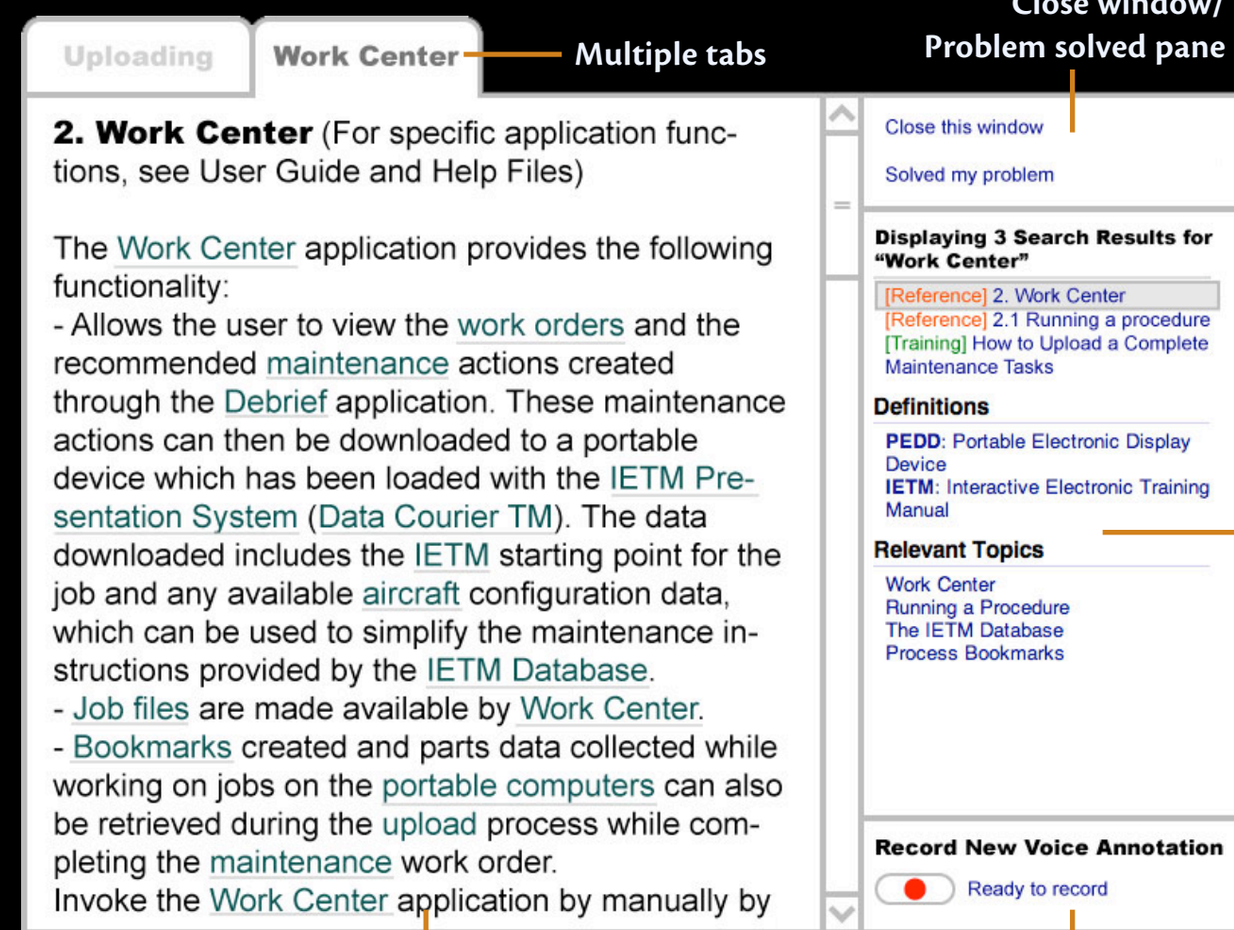
- No small parts that could fall into the plane, get lost, or protrude more than ½ inch from the technician's body
- Ruggedized for high temperatures and harsh environments
- Usable in various lighting conditions
- Usable with or without gloves
- No large, flat screens or keyboards
- Should have a significantly long battery life (at least 4-6 hrs)

Functional Requirements: Software

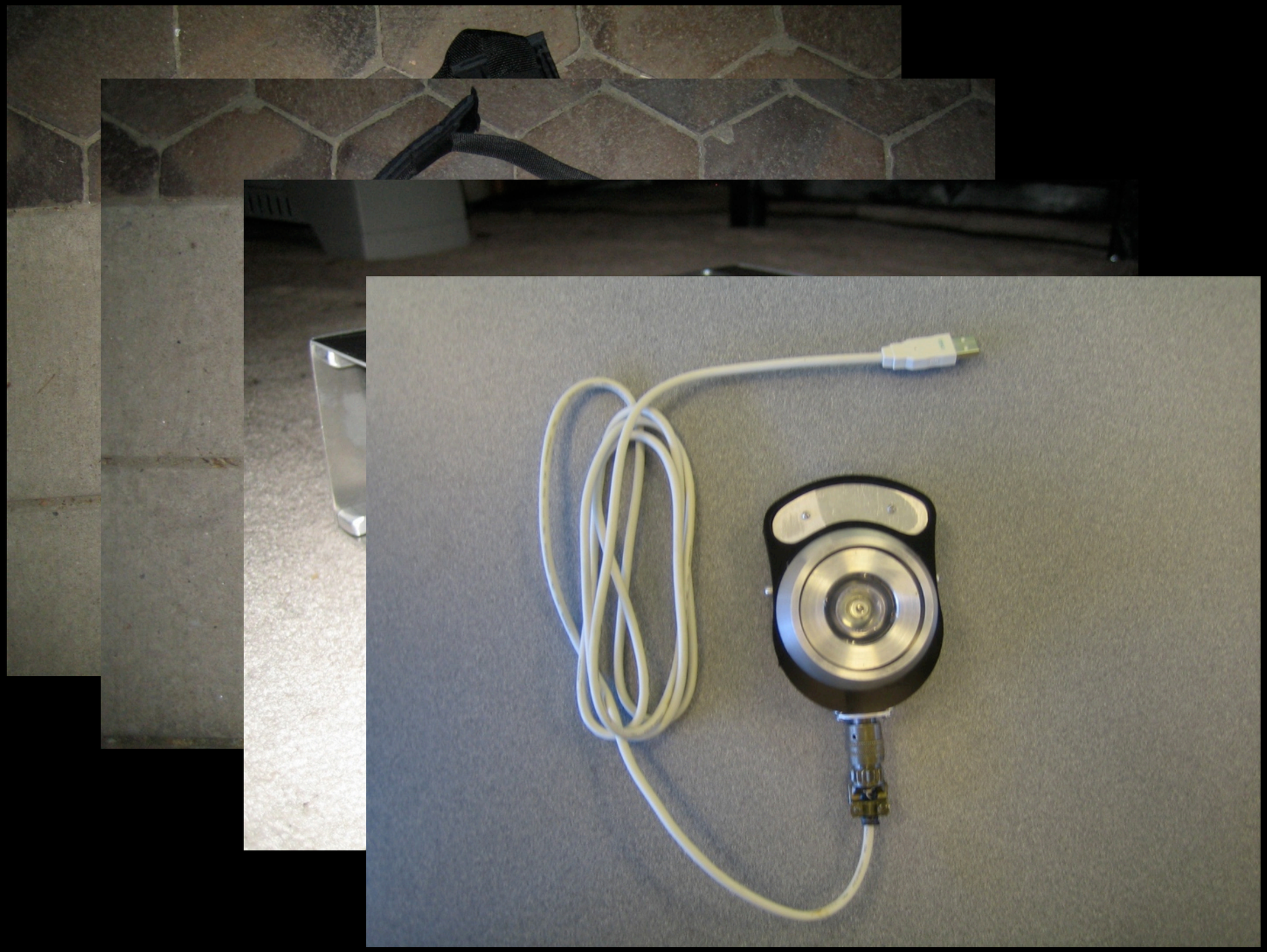
- Should support easy shortcut navigation between the manuals
- Support dynamic links and short cut panes between content
- Capture experience of expert users through annotations
- Manage the software system of manual pages, annotations, database and Web server

System Integration





System Demonstration




(78) 13592554
MADE IN USA

System Demonstration

illoRÉ

Mobile Platform

Mohammad Ahmad
Jules Henry
Megan Hyland
Zack Menegakis
Yong Woo Rhee
Bryon Smith
Adam Wolbach
Daniel Zinzow

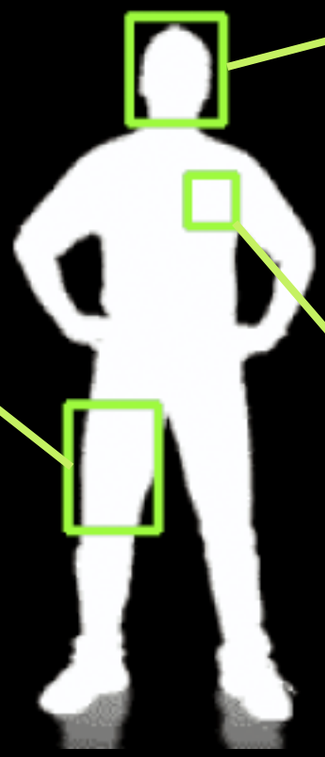
Outline

- Mobile Platform Overview
- Functional Subsystems
  - Dial and Harness
  - Vaio, Enclosure, and Holster
  - Head-Mounted Display and Audio Headset
- Future Work
- Questions

Mobile Platform Goals

Mobile Computer

- Central Processing Unit
- Able to display training manual, web pages, and capture user experience
- Lightweight, durable, dependable, small
- Long Battery Life
- Lightweight



Head-mounted Display, Microphone & Headphones

- Display entire 800x600 SVGA screen
- Voice recording and playback abilities
- Provide ear protection from noise

User Input Device

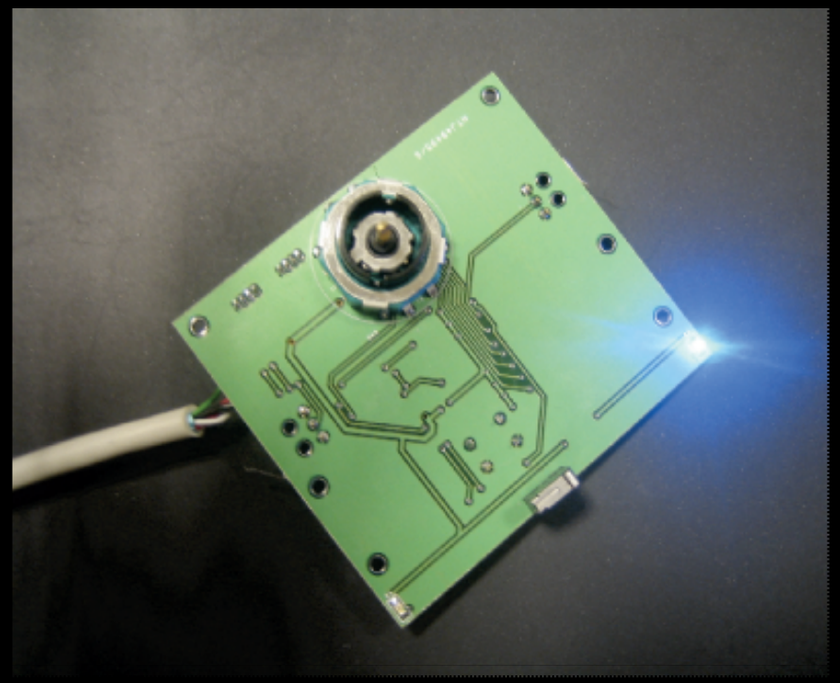
- Point and click capability
- Scrolling/Rotating Functionality
- Auxiliary Buttons
- Mountable for either handedness
- Rugged, reliable
- Used with/without gloves

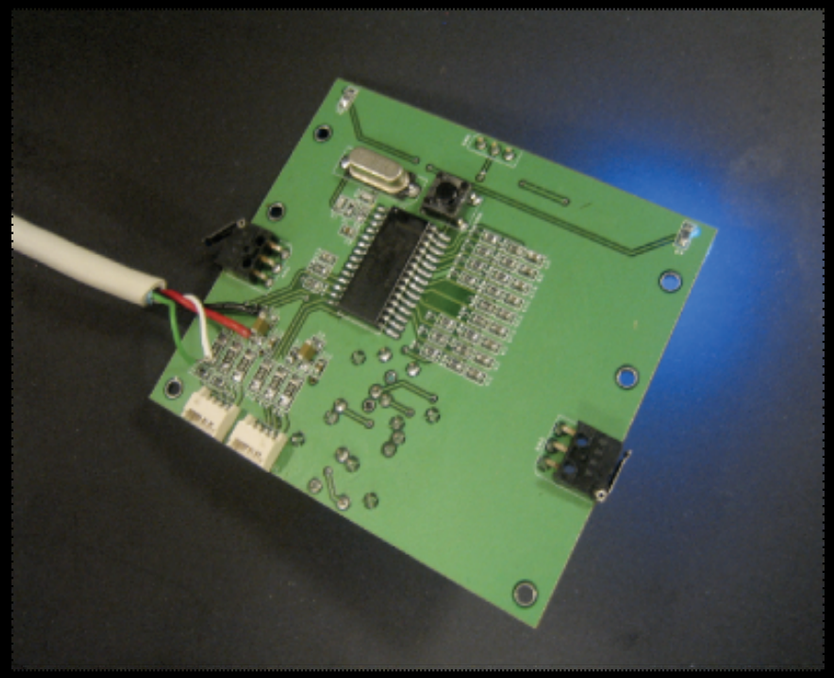
The Dial

- Completed
  - Printed Circuit Board Layout, Manufacturing, Assembly
  - SolidWorks Design
  - On-Campus Machining
  - Software Engineering
  - Integration
- Result: functional input device with a joystick, rotating knob, and four buttons

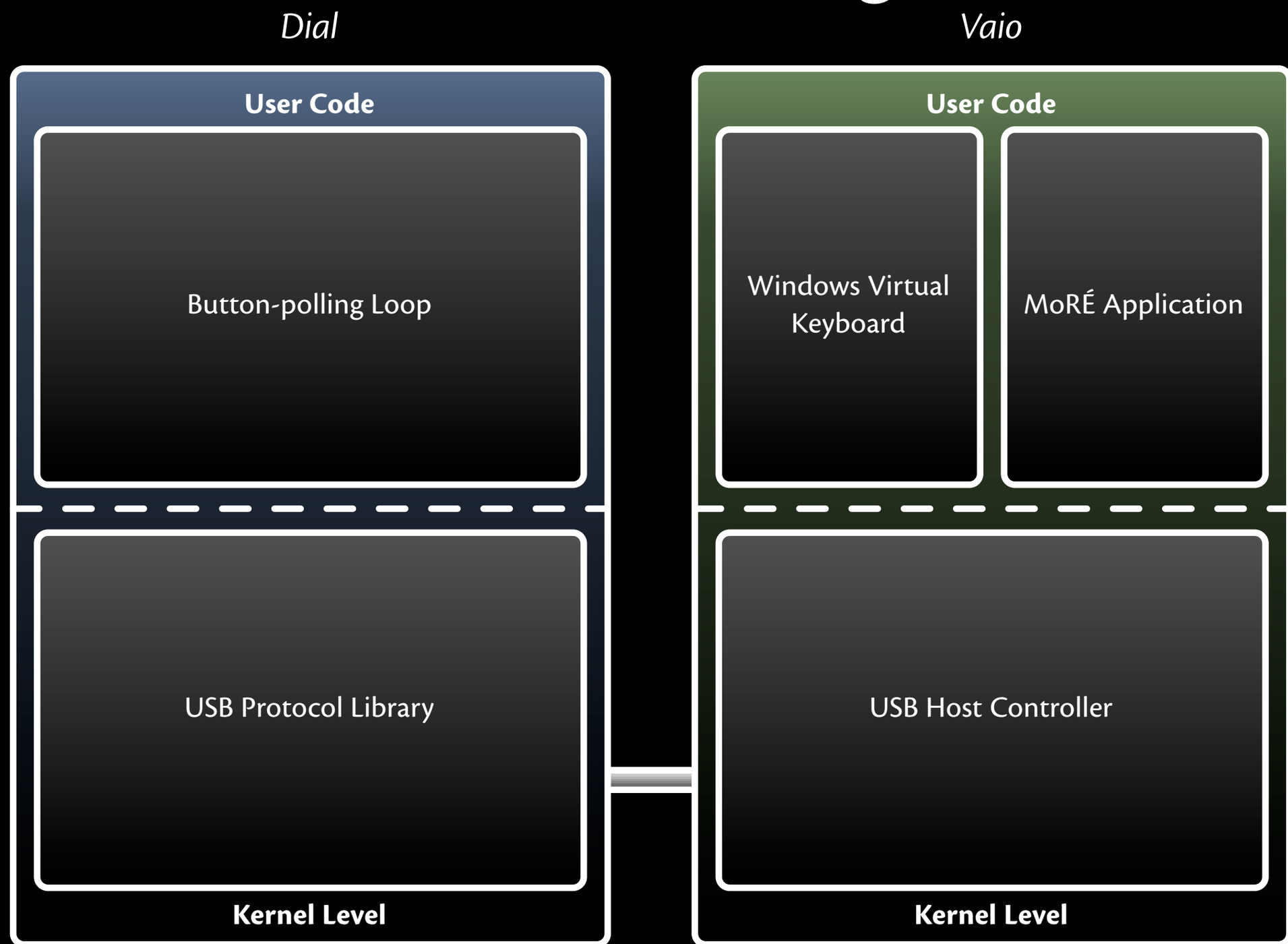
Dial Programming

- Universal Serial Bus (USB) compatible
  - Conforming to the USB Human Interface Device (HID) Class
  - Functions as USB Keyboard
- No host driver needed
- Adapted firmware for another chip found on Freescale's forums to ours
- ~300 new lines of USB firmware code
  - Almost entirely user code





Dial Software Diagram



Dial Harness

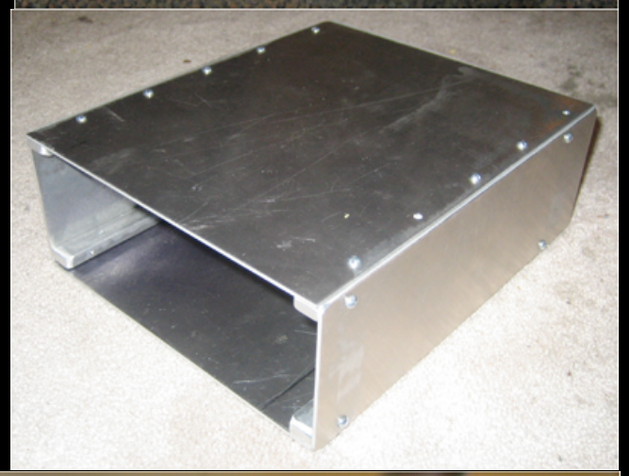
- Resizable buckle design
- Independent of left/right-handedness
- Dial held in place with Velcro

Vaio Holster

- Inspired by Han Solo's holster
- Contains ruggedized aluminum case
- Resizable waist buckle belt
- Thigh strap

Vaio Enclosure

- Ruggedized aluminum case
  - ~0.1" thick
- Size: 7.75" x 6.38" x 2.38"
  - Vaio: 6.6" x 4.3" x 1.5"
- Foam-padded interior
- Fits inside holster
- Good heat conductor



Headset/Head Mounted Display Integration

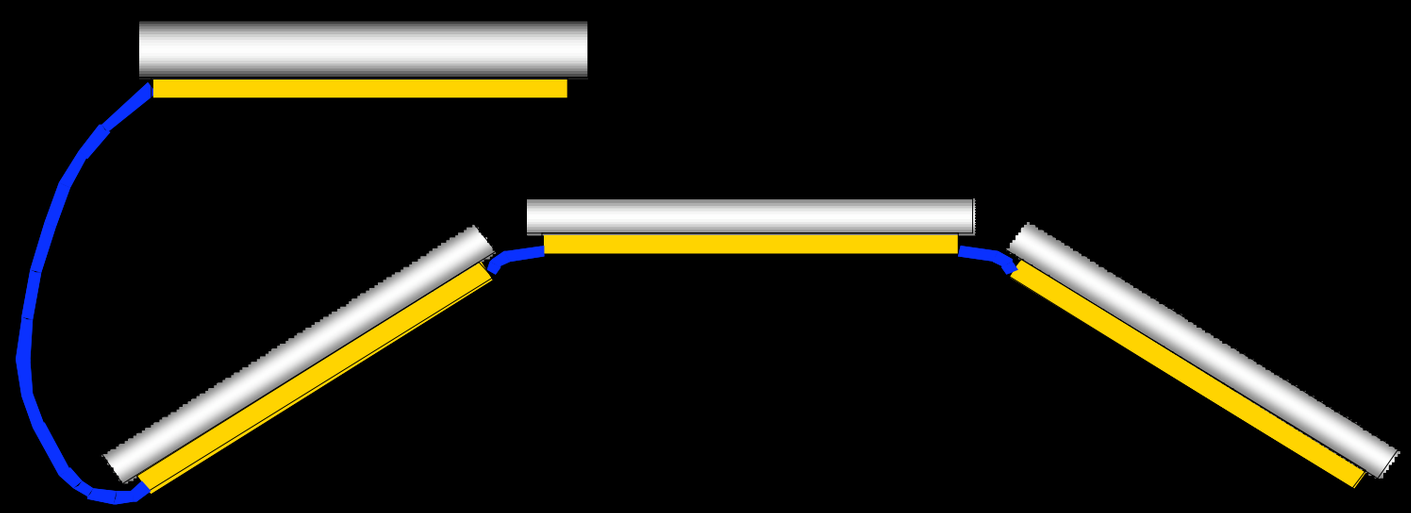
- Right ear conflict visible with both headsets on
- Solved by experimenting; can be screwed together

Integrated System Statistics

- **Combined system weight**: 6.8 pounds
- **System runtime (extended battery)**: 4.9 hours
- **Peak CPU temperature:** 66.0°C
  - Windows XP never reported overheating

Future Work

- Extend battery life via hot-swap controller
  - Software engineering
  - SolidWorks design and machinery of battery clips
  - Belt modifications to hold batteries
  - Create cables/new wiring



illoRÉ

Search

Brian Ellis
Anand Gopalkrishnan
Tabreez Govani
Melanie Haskell
Sachin Kulkarni
Kevin Smith

Software Architecture

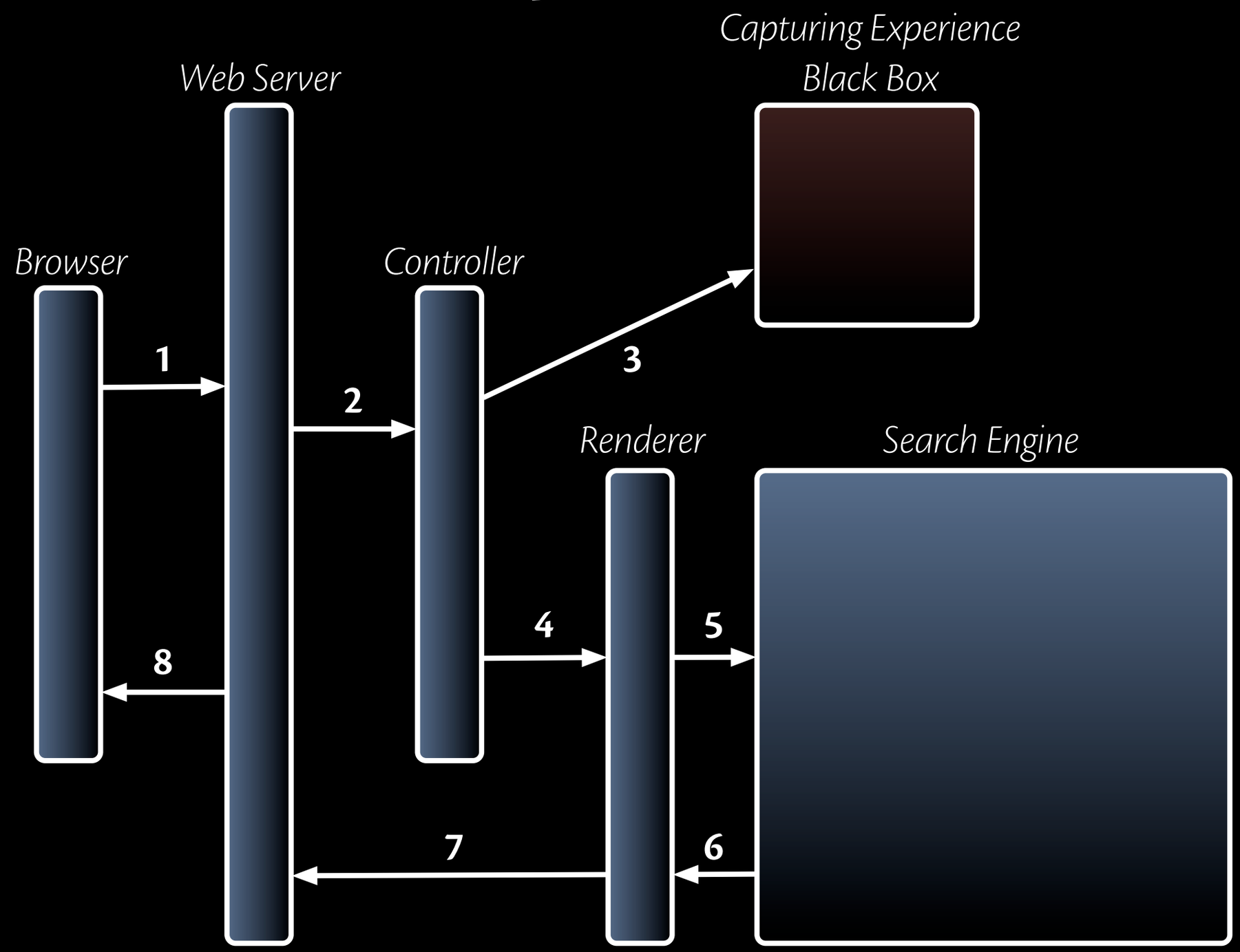
Overview

- Search Team's Responsibilities
- Flowchart
- Search subsections
  - Indexer
  - Controller
  - Search Engine
  - Renderer
  - Proxy
  - Browser
- Future Work/Known Issues

Responsibilities

- Provide a *useful, fast* mechanism for keyword searching through the manuals and collect information from manuals to be leveraged for shortcut navigation
- Cater to the dial input device by *re-rendering* manual pages with smart keyword links and shortcut panes, and implementing a customized browser for navigation
- Allow user to add voice annotations on individual pages which would propagate through the server to all users

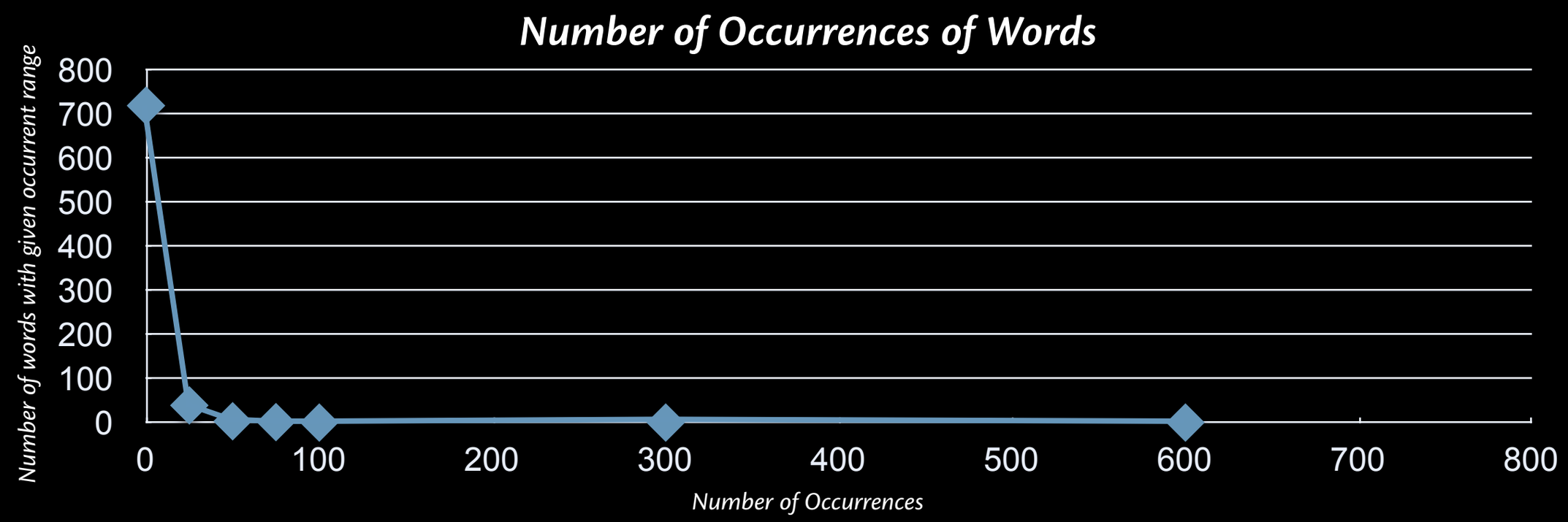
Search Request Flowchart



Indexer

- Parses *both* manuals — XML and HTML
- Uses marks to mark up certain tags for better search
  - **Bold, strong, title:** important
  - **Javascript:** instructions
  - Keeps track of offset into page (proximity)
- Adds index, keywords, pages, titles to database
- Differentiates between common English words in the tail distribution to avoid adding them as keywords
- Highly configurable
  - Easy to change what tags are important
  - Easy to specify actions for tags down to the attributes and their values

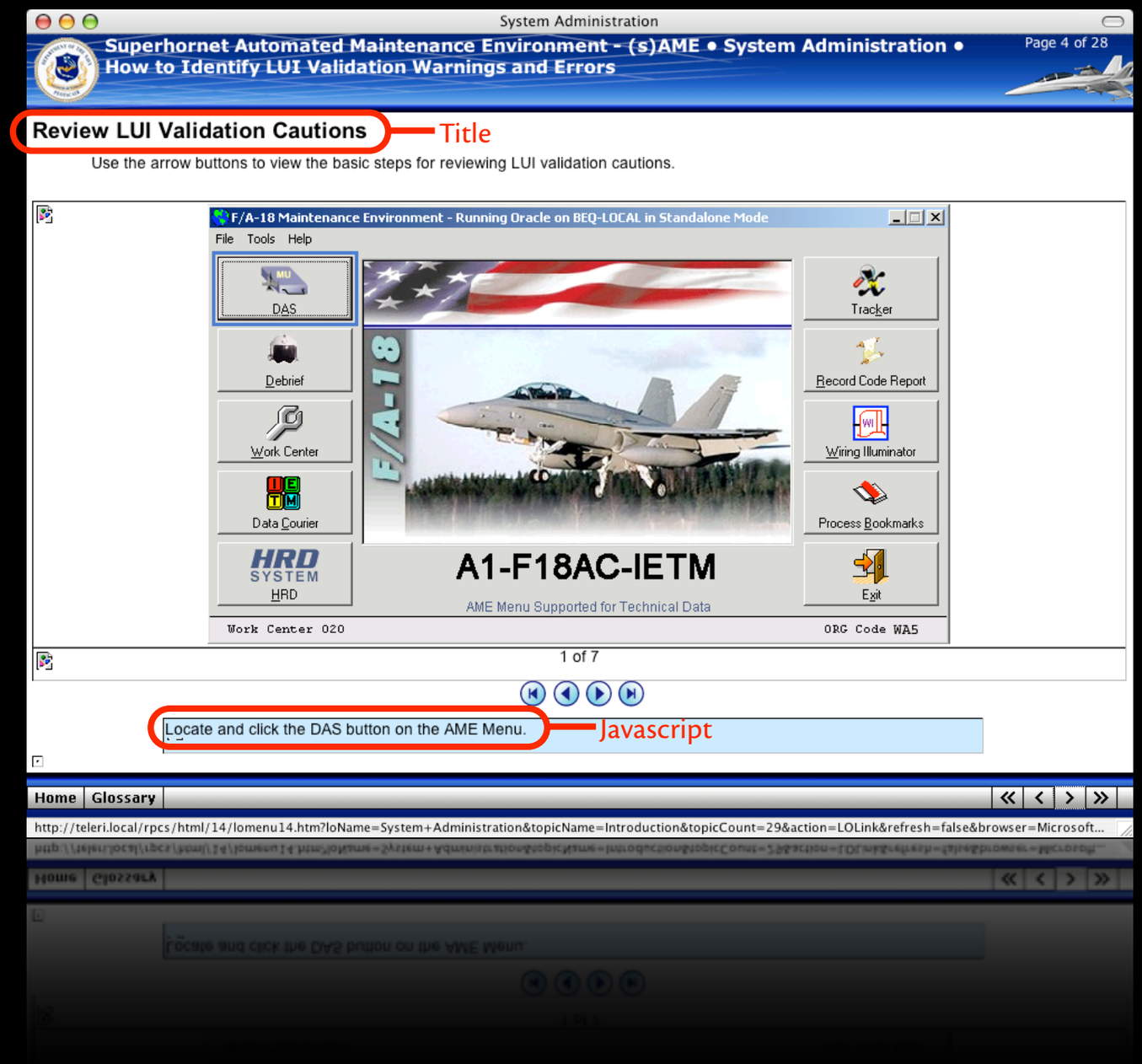
Indexer: Word Statistics



- Statistics based on 81 manual pages indexed
- **7,398** total words
- **786** unique words
- 1st place: “the” (**604**)
- 2nd place: “maintenance” (**318**)
- **234** words only appear once so far
- **1,040** total keywords

Indexer: Formatting Statistics

- **102** titles
- **732** *strong* words (**113** unique)
- **729** *Javascript* words (**228** unique)



Controller

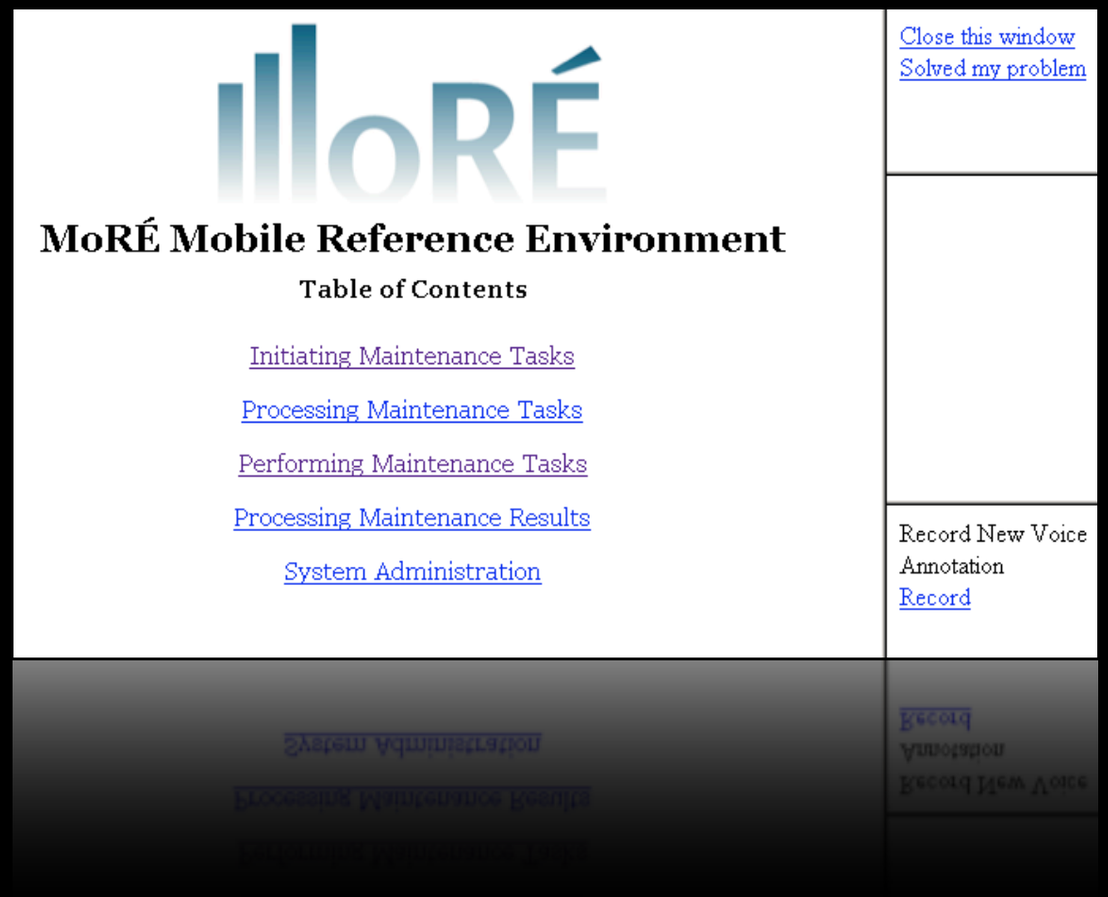
- Distinguishes between the different user actions (search, login, etc.)
- Allows users to log in via a JSP (Java Servlet Pages) page
  - Stores a session of the user as a Java bean so that information about the user can be accessed at any time
- Includes a wrapper class for database access, queries, etc.

The Engine

- Performs multiple keyword search
- Ranking — important for "useful" goal
  - Uses style information
  - Uses location information
  - Prefers pages with the most words matched
- Speed: complex comparisons and lookups all in the database since the database is highly tuned to execute these types of queries
- Highly configurable by adjusting ranking parameters

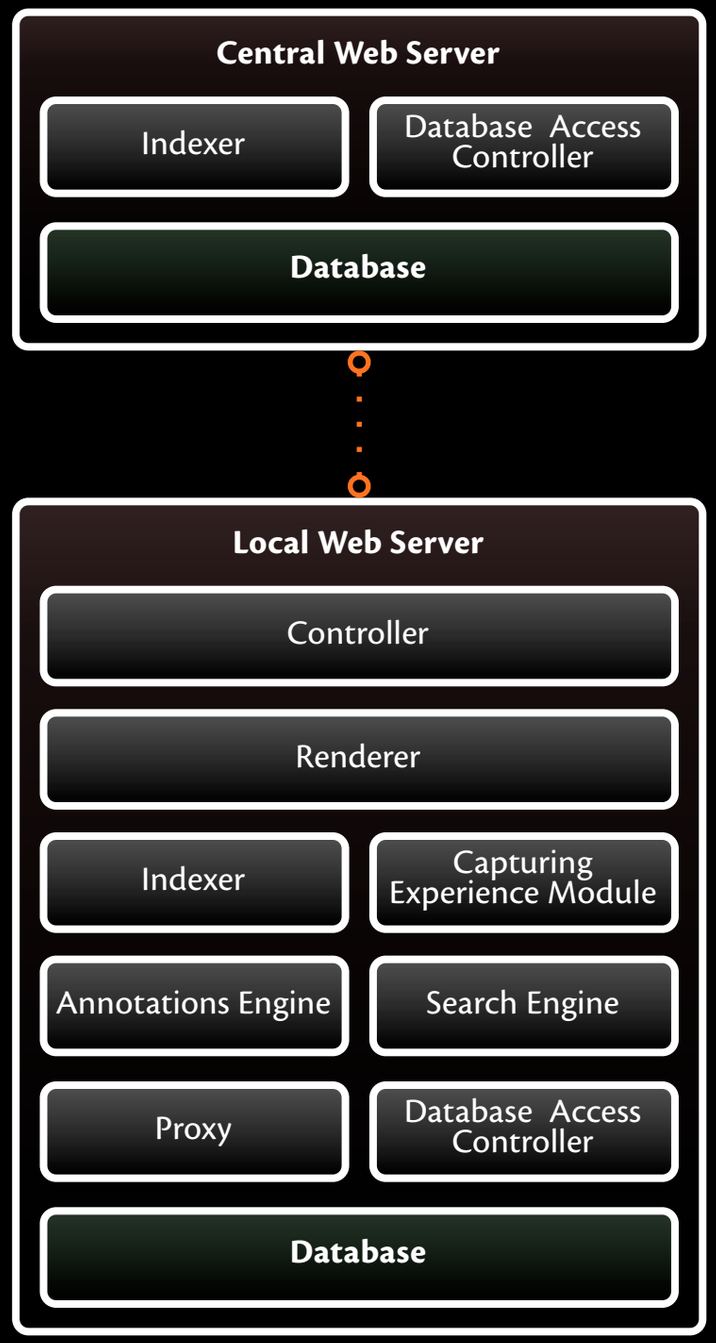
Renderer

- Dynamically builds pages
- Adds links for keywords in both manuals
  - Avoids adding code in HTML tags, Javascript, and CSS
- Displays three menu panes on the right
  - Navigation pane: allows user to close a window and indicate that a page was helpful for solving a problem (used by capturing experience)
  - Link pane: shows search results as well as static and dynamic smart links
  - Annotations pane: allows user to record annotations
- Modifies content resolution and image/object sizes for easy viewing on the HUD



Updates: The Proxy

- Central/client database:
  - Manually run update synchronizes the two databases
  - Binary log comparison used to minimize bandwidth used when updating
- Central file access:
  - Each request looks locally for the file; if not found, downloads it from the central server; then displays it



Customized Browser

- Rationale:
  - Events from the dial input device must be handled
  - Must support custom navigation like switching panes and tabs
- Implementation:
  - Created using XML User Interface Language (XUL) with the Mozilla browser
  - Handles dial events to navigate through pages
  - Allows rendered pages to appear properly
  - Usable with any standard input device (mouse, keyboard)
  - Easy to install and runs on virtually any machine

Future Work/Known Issues

- Indexer:
  - User relevant keywords
  - Generate title from relevant text when not supplied
  - Phrase indexing
- Search engine:
  - Allow for phrase searching
  - Make results more heavily influenced by experience capture data
- Web server/Proxy:
  - Automated client and central database sync
  - Two-way sync so annotations propagate to all clients
- Renderer:
  - Improved resizing algorithm for images and objects
- Browser
  - Improved image resizing/ compression algorithm (actually a flaw with Mozilla itself)

illoRÉ

Experience Capture

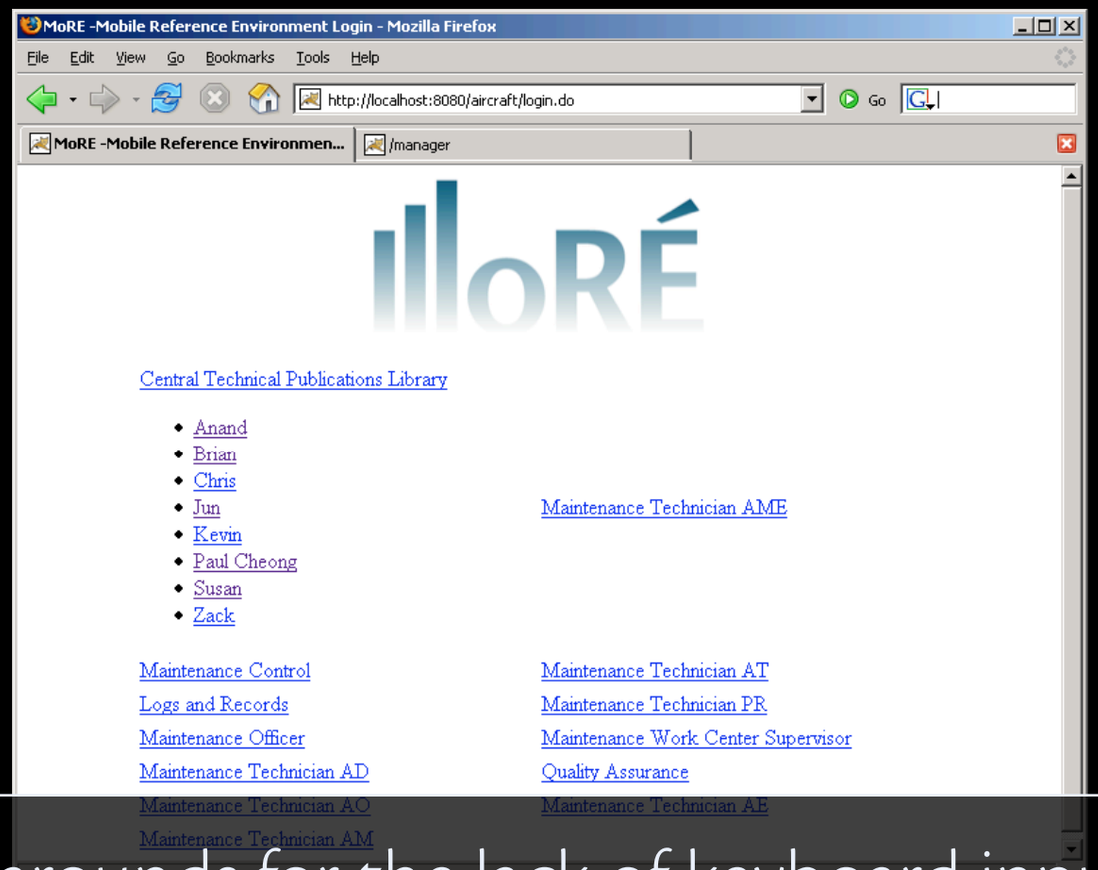
Jae Ha
Leon Mai
Aashni Shah
Wen Shu Tang
Jonathan Tran

Overview

- Goals:
  - To capture the experience of users and help future users
  - To recognize when a user needs assistance
- Tools devised:
  - Annotations
  - Smart links
  - Cycling detection

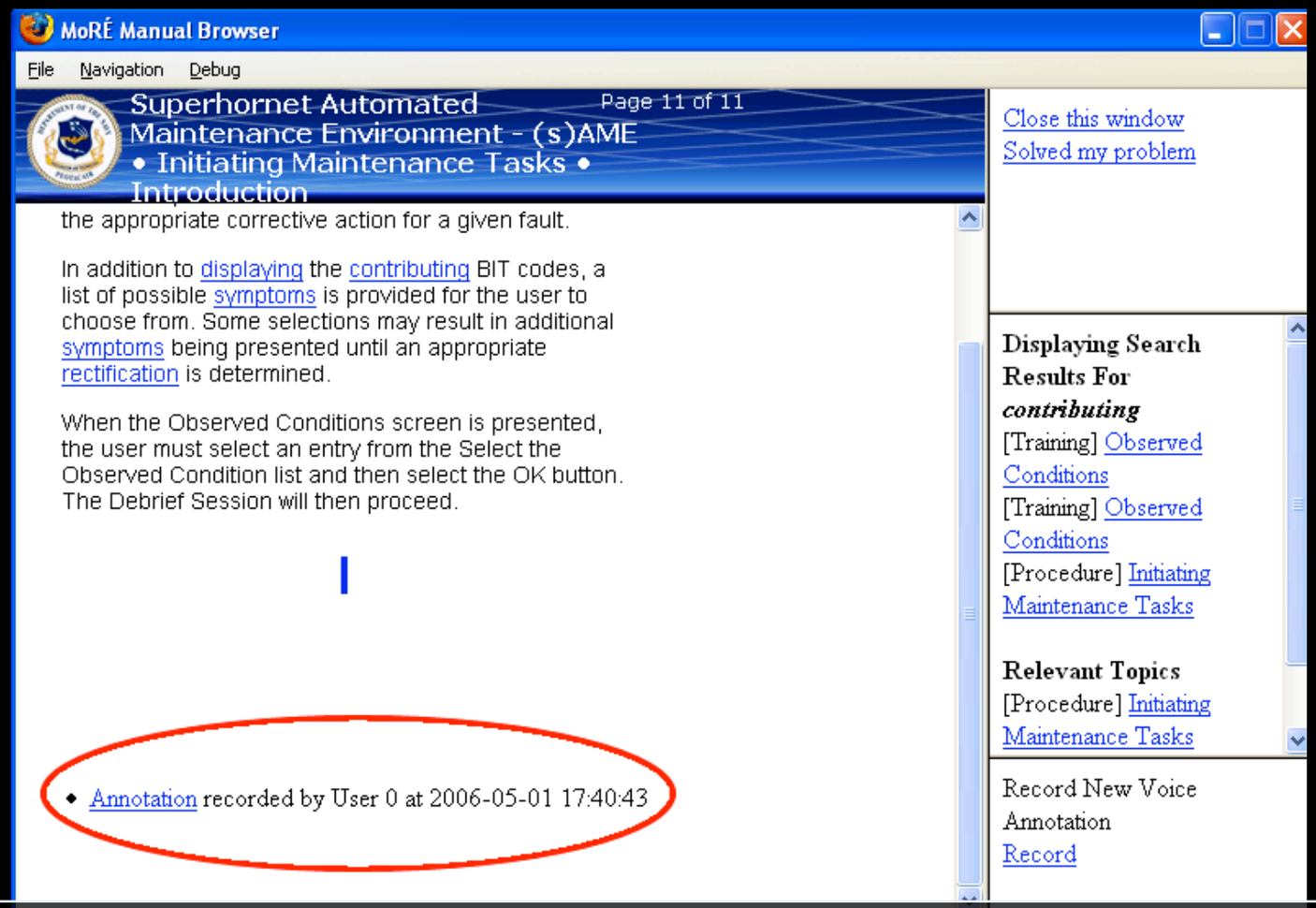
Software Architecture

Login



- Links are workarounds for the lack of keyboard input
- Alternatively, could be position names instead of user names — e.g., "maintenance technician"
- Assume trusted users — no passwords

Annotations



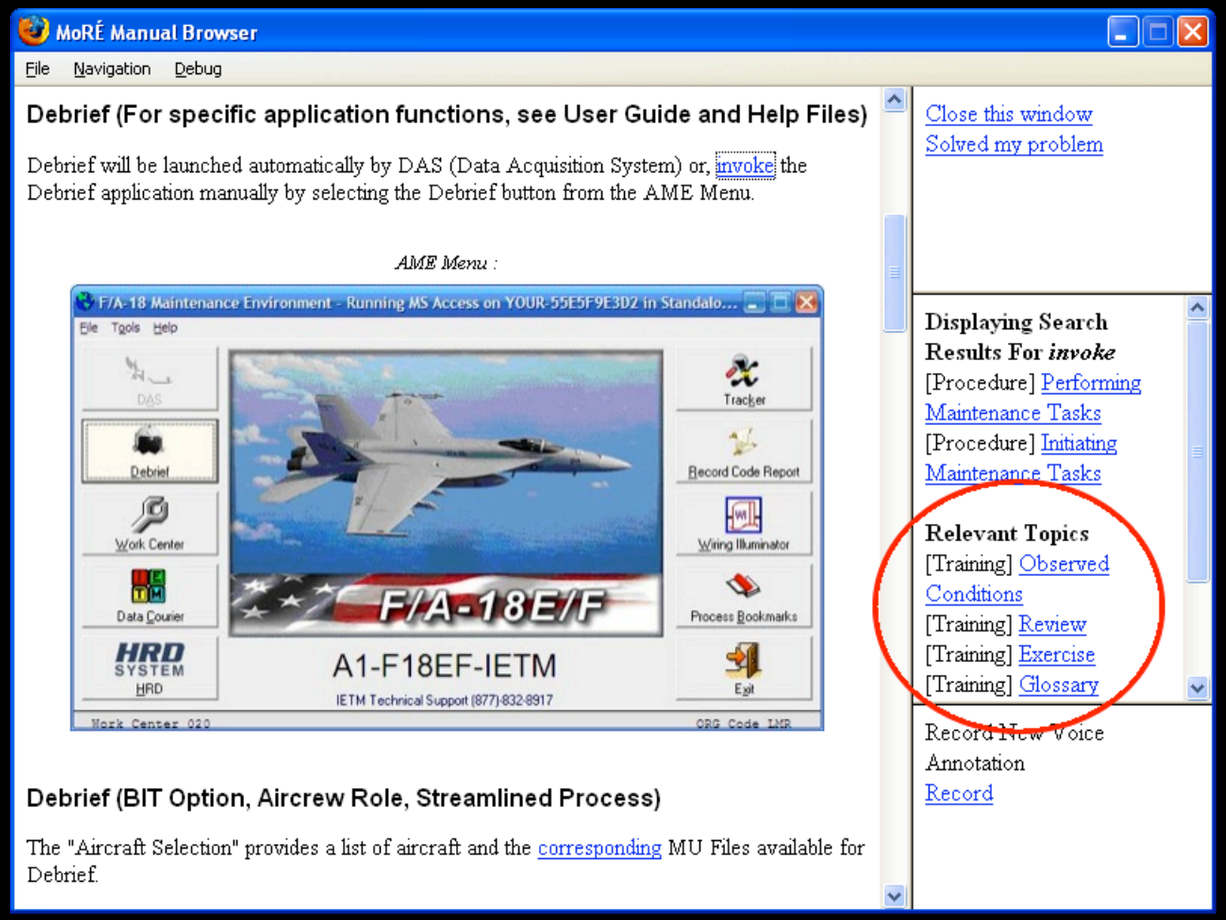
- Audio annotations can be left by users
- Displayed at the bottom of content in the browser

Annotations

- Features:
  - Button in custom browser to start and stop audio recording
  - Variable length recordings
- Captured audio is saved in GSM format (Global System for Mobile telecommunication)
  - About ten times smaller than raw WAV audio
  - Quality is good for voice (and music)
- Audio files and metadata stored locally
  - Shows user who left the annotation, date left
  - User is identified by login

Smart Links

- Goal: Link to other relevant pages
  - Bypass extra steps
  - Find related info without searching
- Two kinds of smart links:
  - Statically generated
  - Dynamically generated



Static Smart Links

- In reference manual, show users the relevant training manual topics
- Automatically generated at index time
- Implementation overview:
  1. Extract headings from reference manual
  2. Search the training manual
  3. Display best results on reference manual pages
- Smart links can also point within the same manual

Dynamic Smart Links

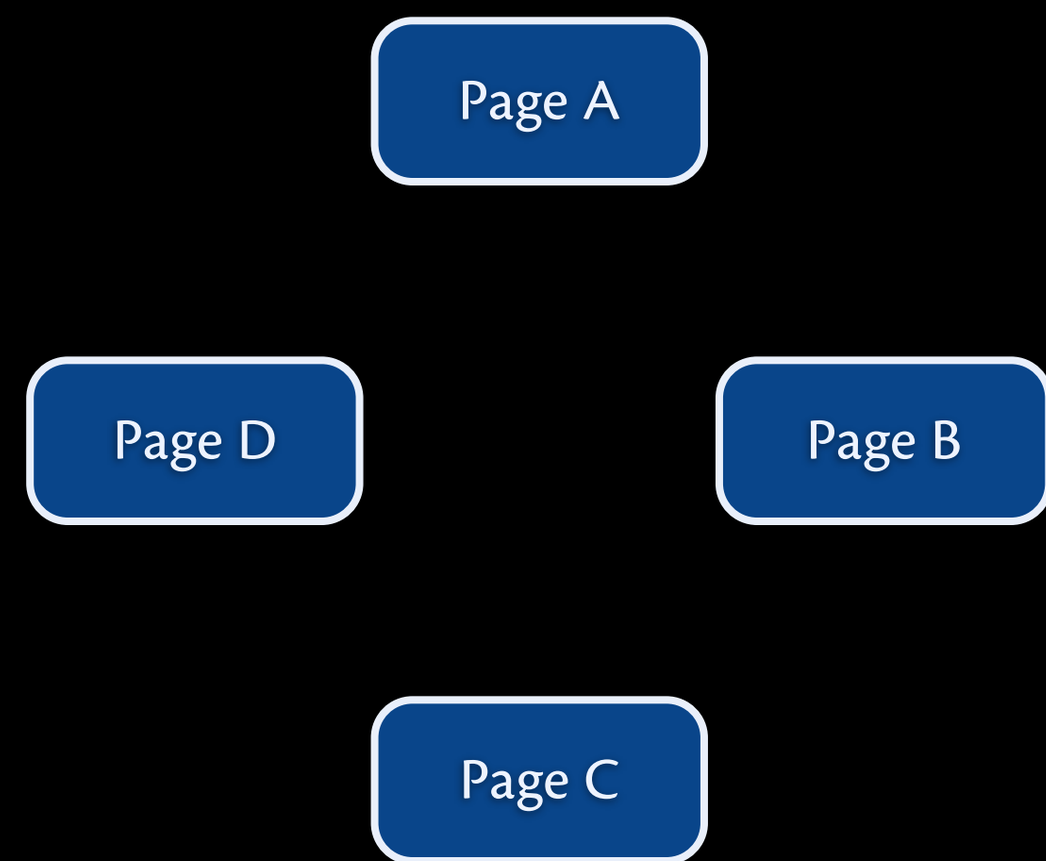
- Allow usage to generate new links
  - Learn from other users
- Semi-automatic
  - Users specify when they find what they're looking for
- Context sensitive
  - Suggestions based on topic being viewed
- Exploits transitivity
  - If topic B was relevant to topic A, and topic C was relevant to topic B, then topic C is probably relevant to topic A
  - A → B and B → C, so A → C

Dynamic Smart Links: Detailed Algorithm

1. Pages viewed are tracked
2. When solution is found, edges are made from all pages to the solution page
3. Process is repeated with other pages, and weights are increased
4. Upon future views, pages link to the highest ranked pages they are connected to

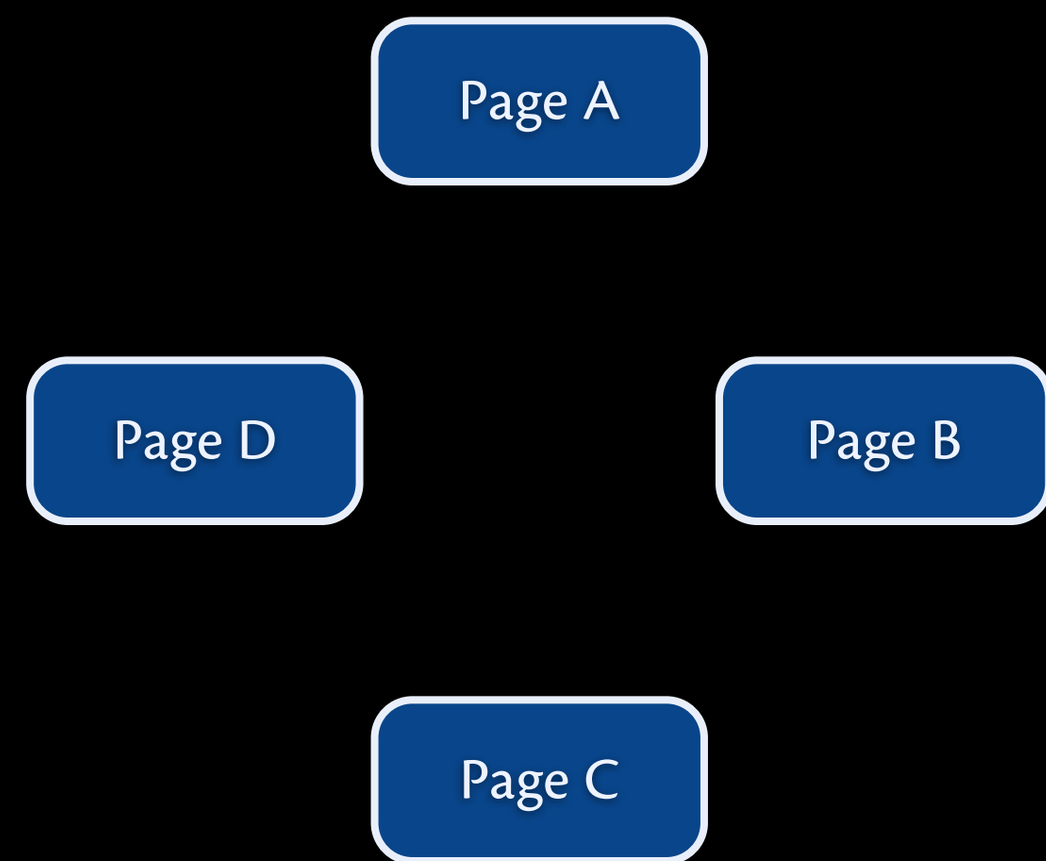
Dynamic Smart Links: Detailed Algorithm

1. Pages viewed are tracked
2. When solution is found, edges are made from all pages to the solution page
3. Process is repeated with other pages, and weights are increased
4. Upon future views, pages link to the highest ranked pages they are connected to



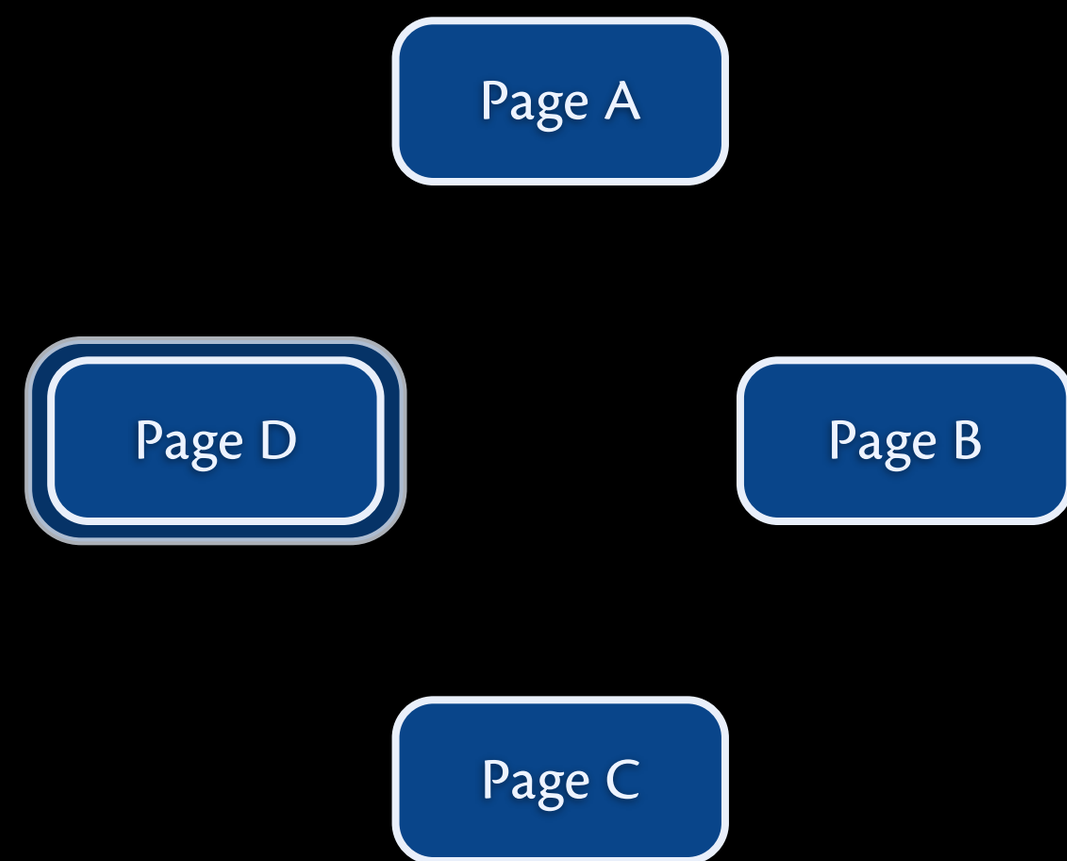
Dynamic Smart Links: Detailed Algorithm

1. Pages viewed are tracked
2. When solution is found, edges are made from all pages to the solution page
3. Process is repeated with other pages, and weights are increased
4. Upon future views, pages link to the highest ranked pages they are connected to



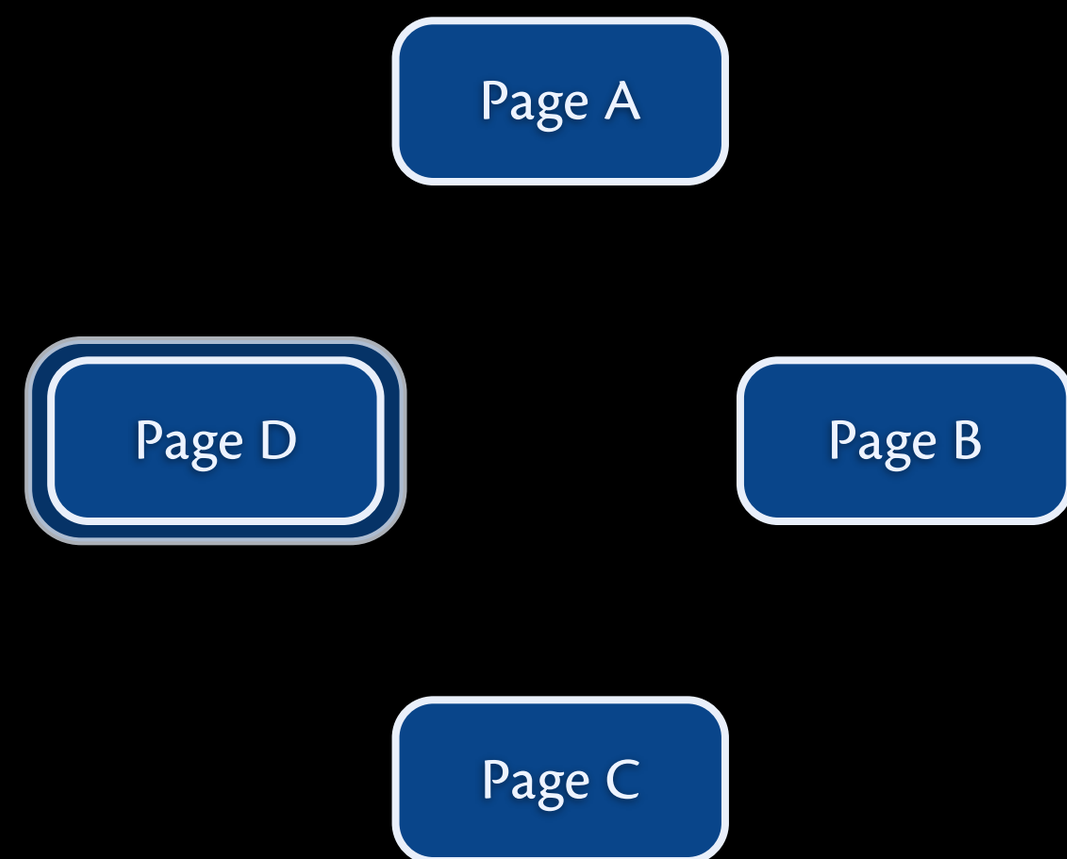
Dynamic Smart Links: Detailed Algorithm

1. Pages viewed are tracked
2. When solution is found, edges are made from all pages to the solution page
3. Process is repeated with other pages, and weights are increased
4. Upon future views, pages link to the highest ranked pages they are connected to



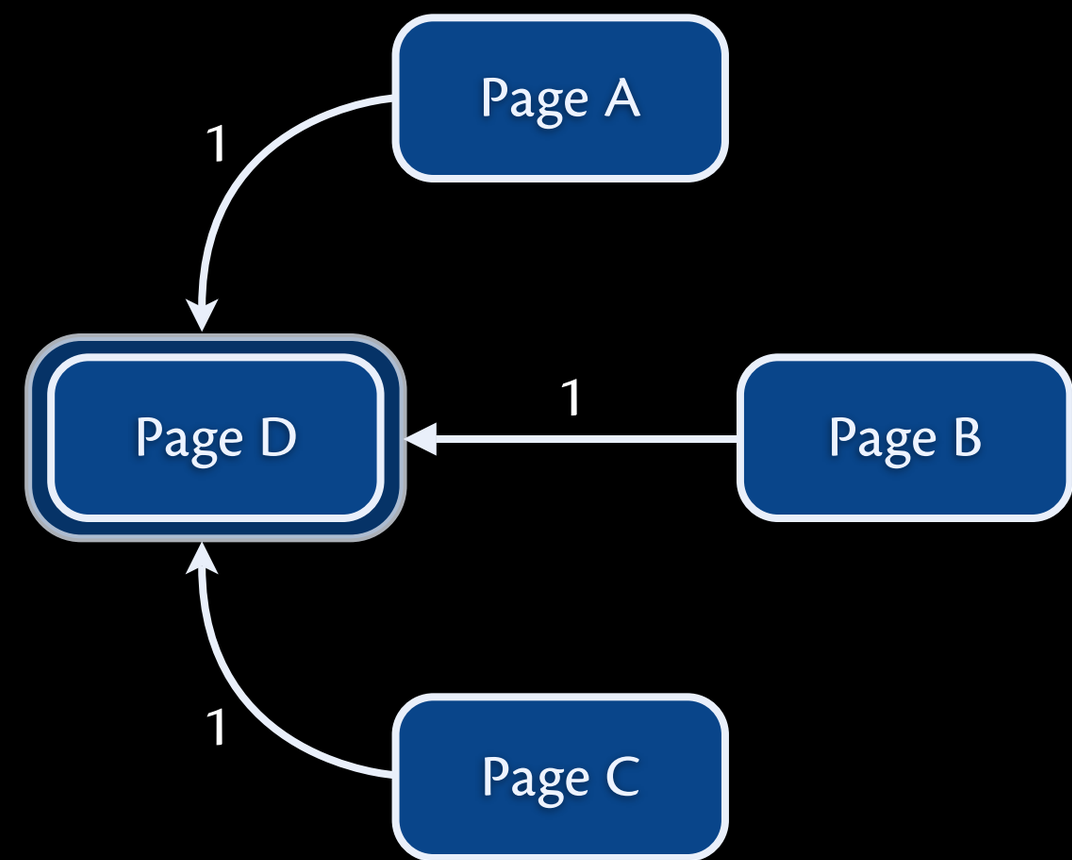
Dynamic Smart Links: Detailed Algorithm

1. Pages viewed are tracked
2. When solution is found, edges are made from all pages to the solution page
3. Process is repeated with other pages, and weights are increased
4. Upon future views, pages link to the highest ranked pages they are connected to



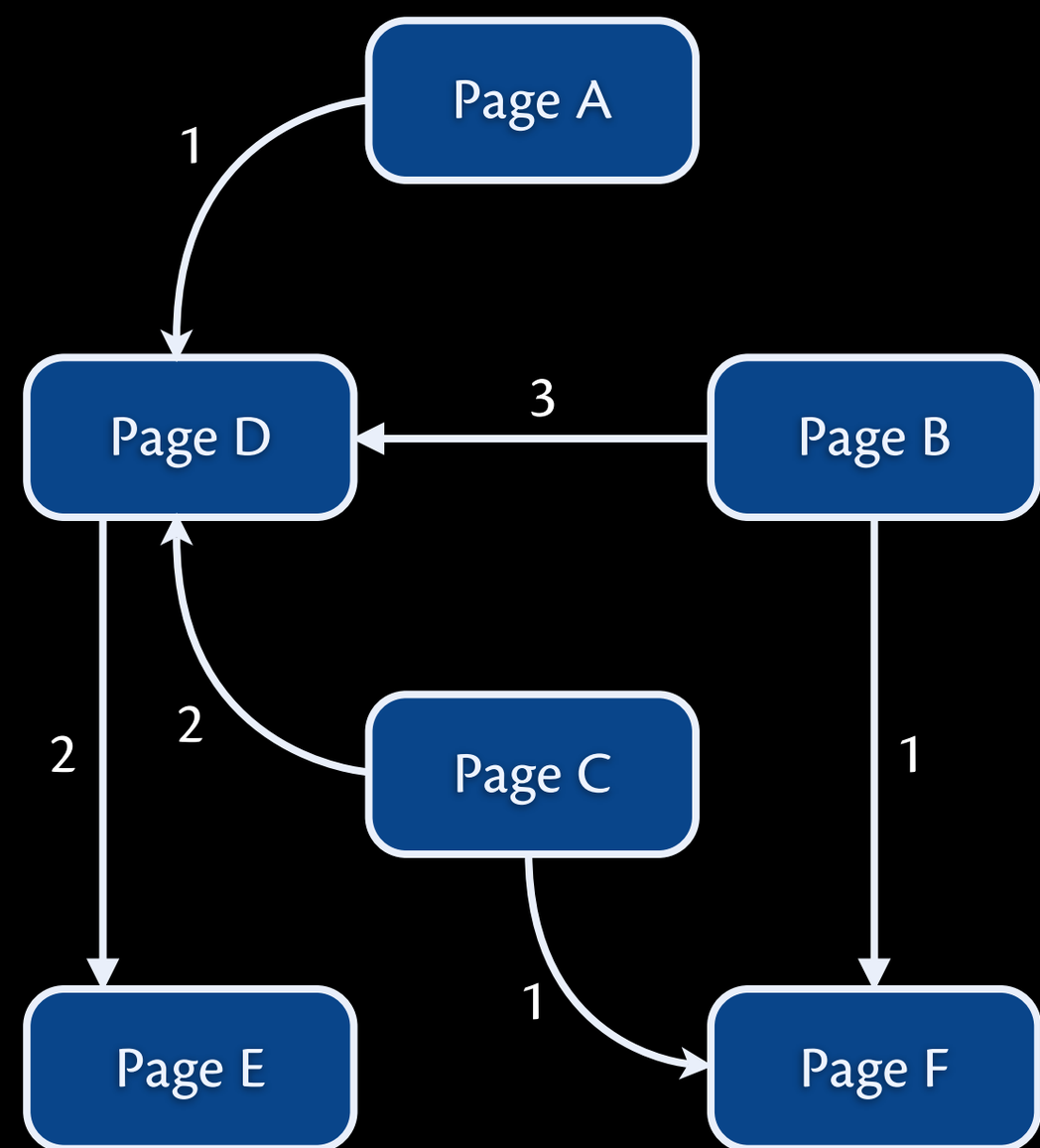
Dynamic Smart Links: Detailed Algorithm

1. Pages viewed are tracked
2. When solution is found, edges are made from all pages to the solution page
3. Process is repeated with other pages, and weights are increased
4. Upon future views, pages link to the highest ranked pages they are connected to



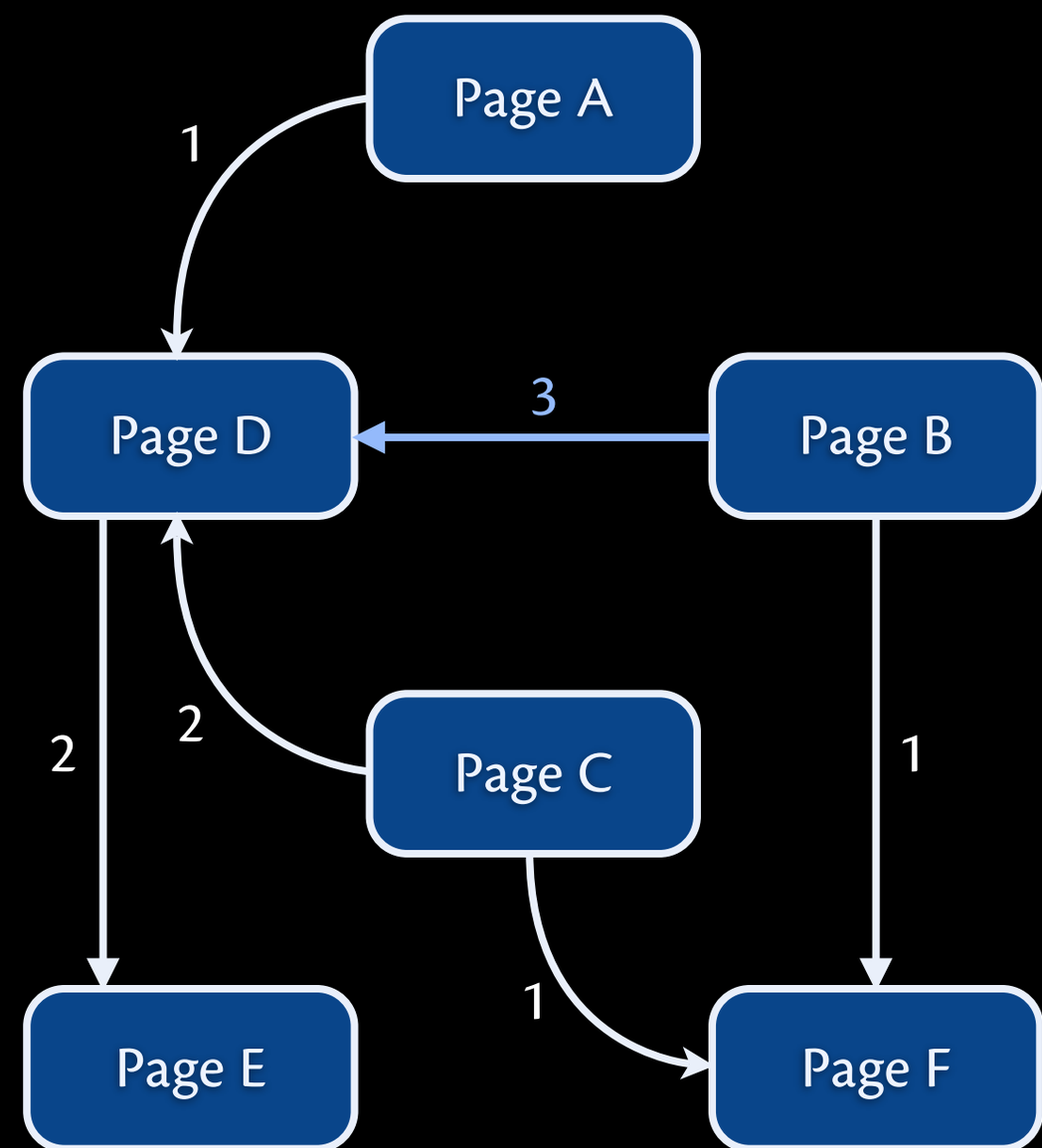
Dynamic Smart Links: Detailed Algorithm

1. Pages viewed are tracked
2. When solution is found, edges are made from all pages to the solution page
3. Process is repeated with other pages, and weights are increased
4. Upon future views, pages link to the highest ranked pages they are connected to



Dynamic Smart Links: Detailed Algorithm

1. Pages viewed are tracked
2. When solution is found, edges are made from all pages to the solution page
3. Process is repeated with other pages, and weights are increased
4. Upon future views, pages link to the highest ranked pages they are connected to



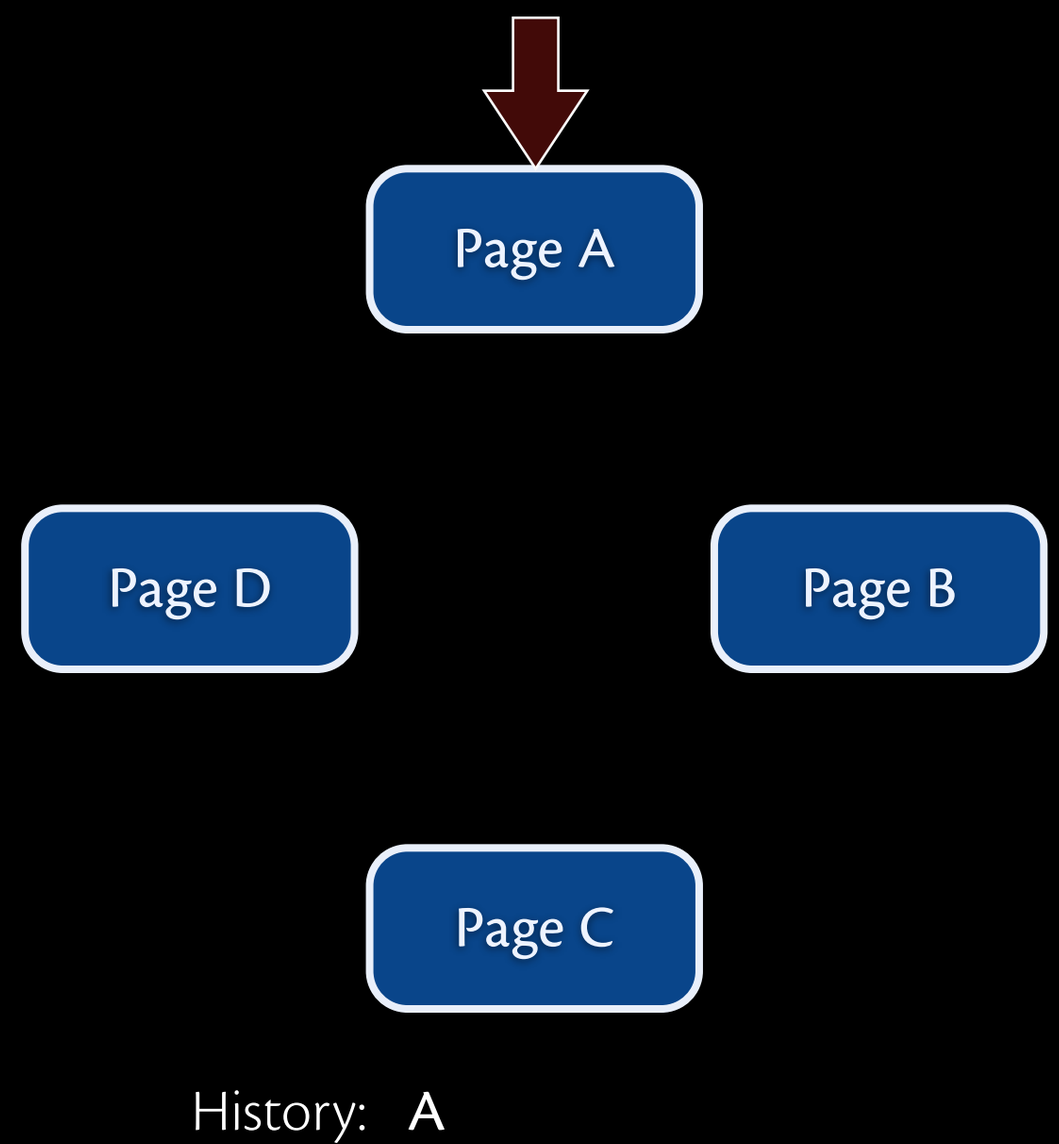
Cycling Detection

- Goal: recognize when users need help
- If user switches between viewing pages such that…
  - one page has been viewed at least three times and
  - another page has been viewed at least twice
- …then the user is probably looking for something and needs help
- Especially the case without browser "back" and "forward" buttons

History:

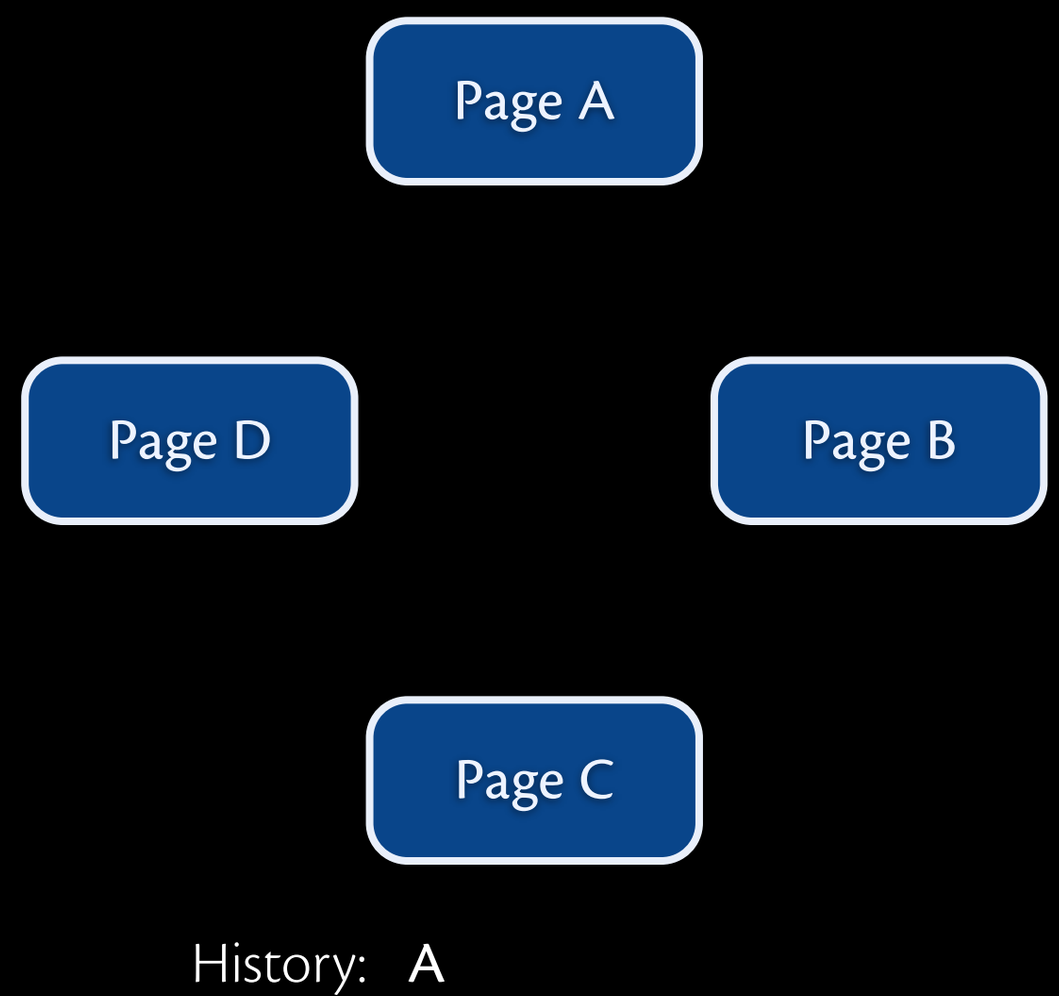
Cycling Detection

- Goal: recognize when users need help
- If user switches between viewing pages such that…
  - one page has been viewed at least three times and
  - another page has been viewed at least twice
- …then the user is probably looking for something and needs help
- Especially the case without browser "back" and "forward" buttons



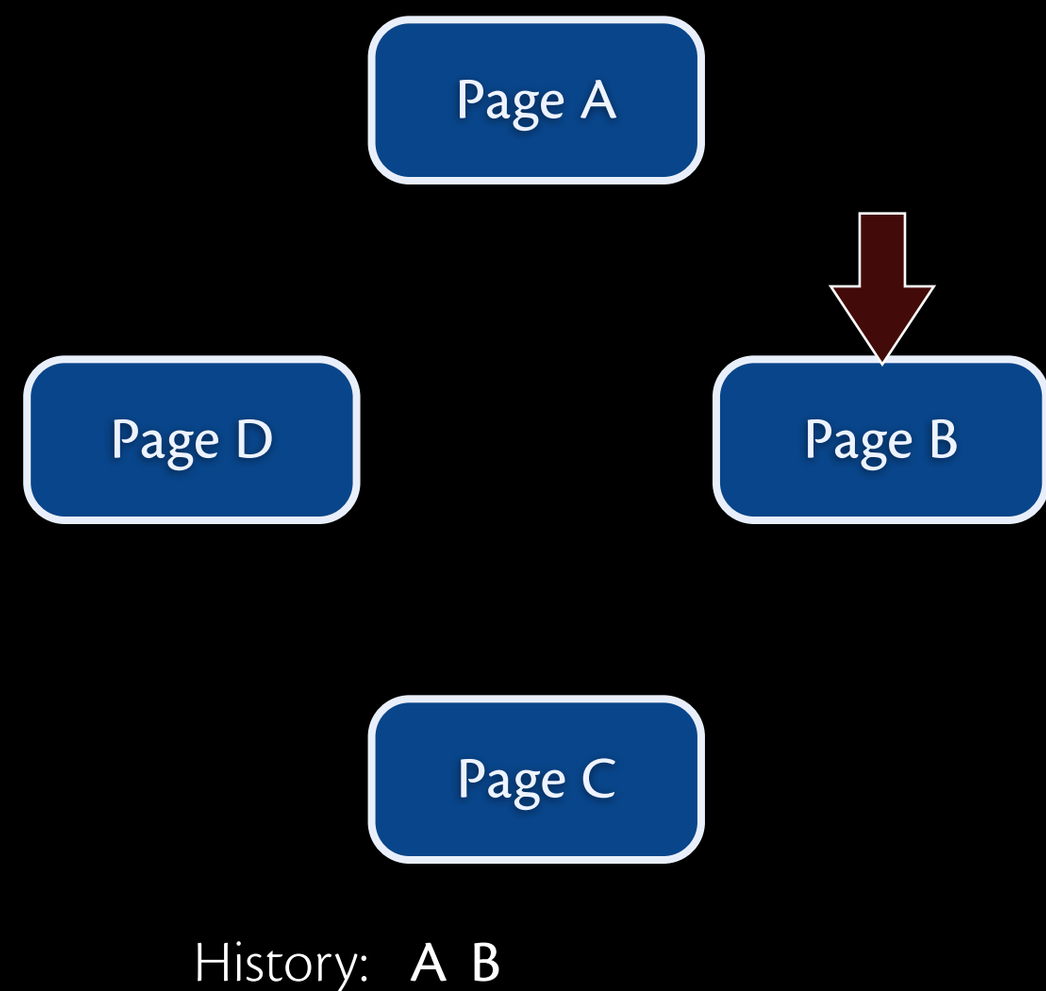
Cycling Detection

- Goal: recognize when users need help
- If user switches between viewing pages such that…
  - one page has been viewed at least three times and
  - another page has been viewed at least twice
- …then the user is probably looking for something and needs help
- Especially the case without browser "back" and "forward" buttons



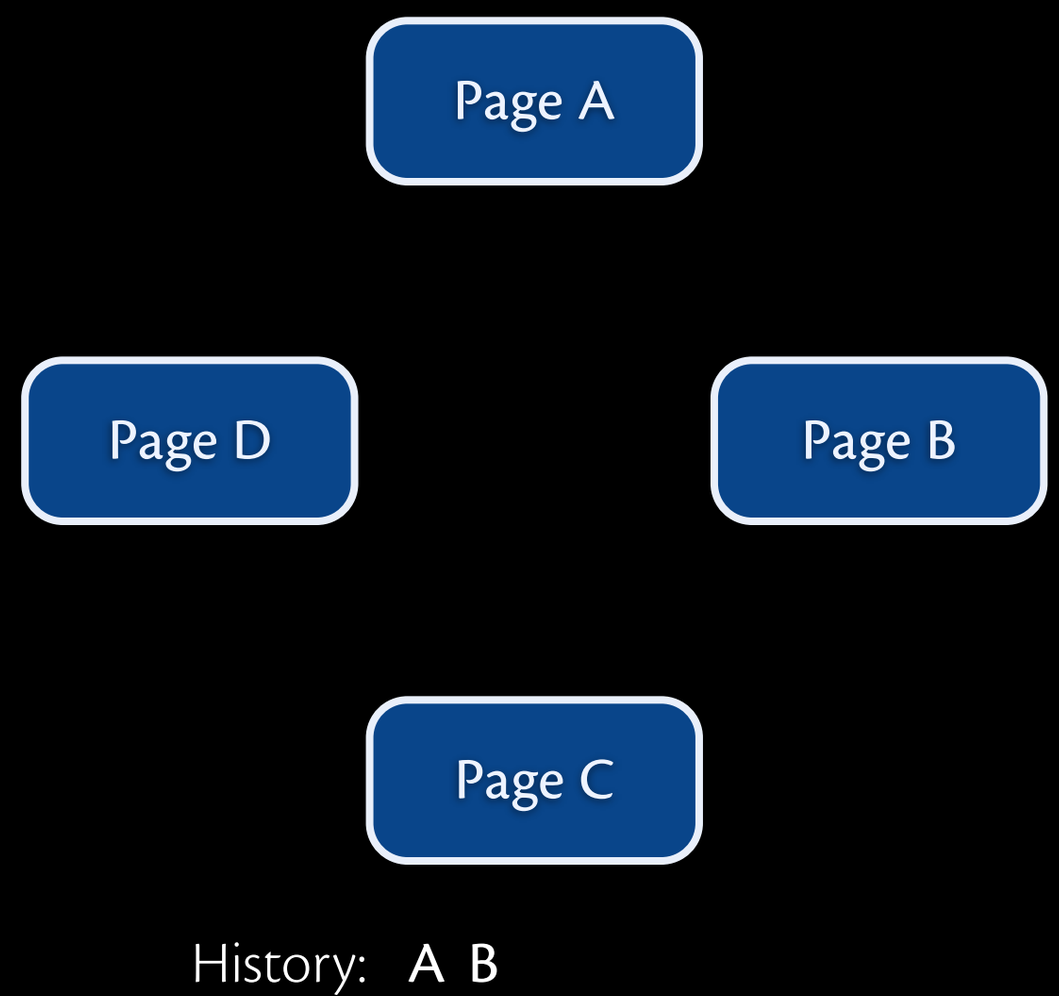
Cycling Detection

- Goal: recognize when users need help
- If user switches between viewing pages such that…
  - one page has been viewed at least three times and
  - another page has been viewed at least twice
- …then the user is probably looking for something and needs help
- Especially the case without browser "back" and "forward" buttons



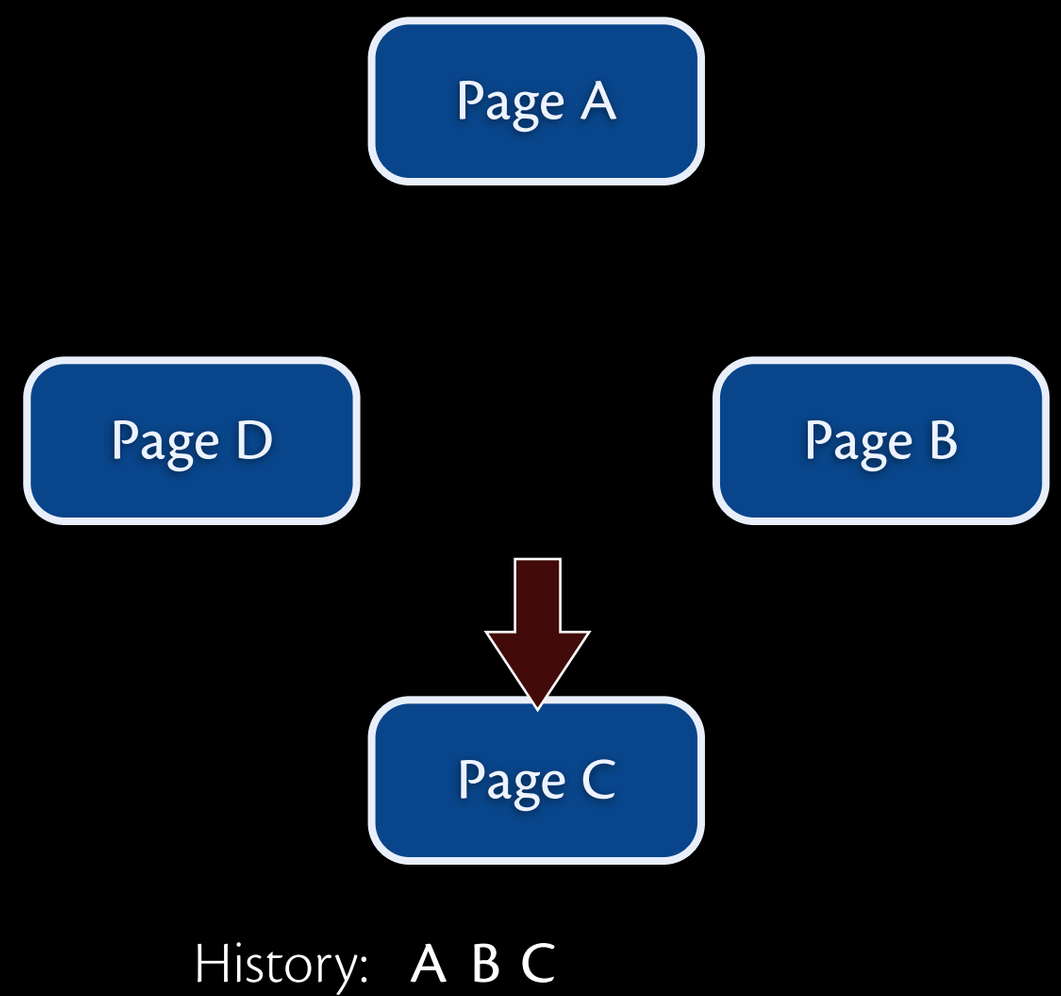
Cycling Detection

- Goal: recognize when users need help
- If user switches between viewing pages such that…
  - one page has been viewed at least three times and
  - another page has been viewed at least twice
- …then the user is probably looking for something and needs help
- Especially the case without browser "back" and "forward" buttons



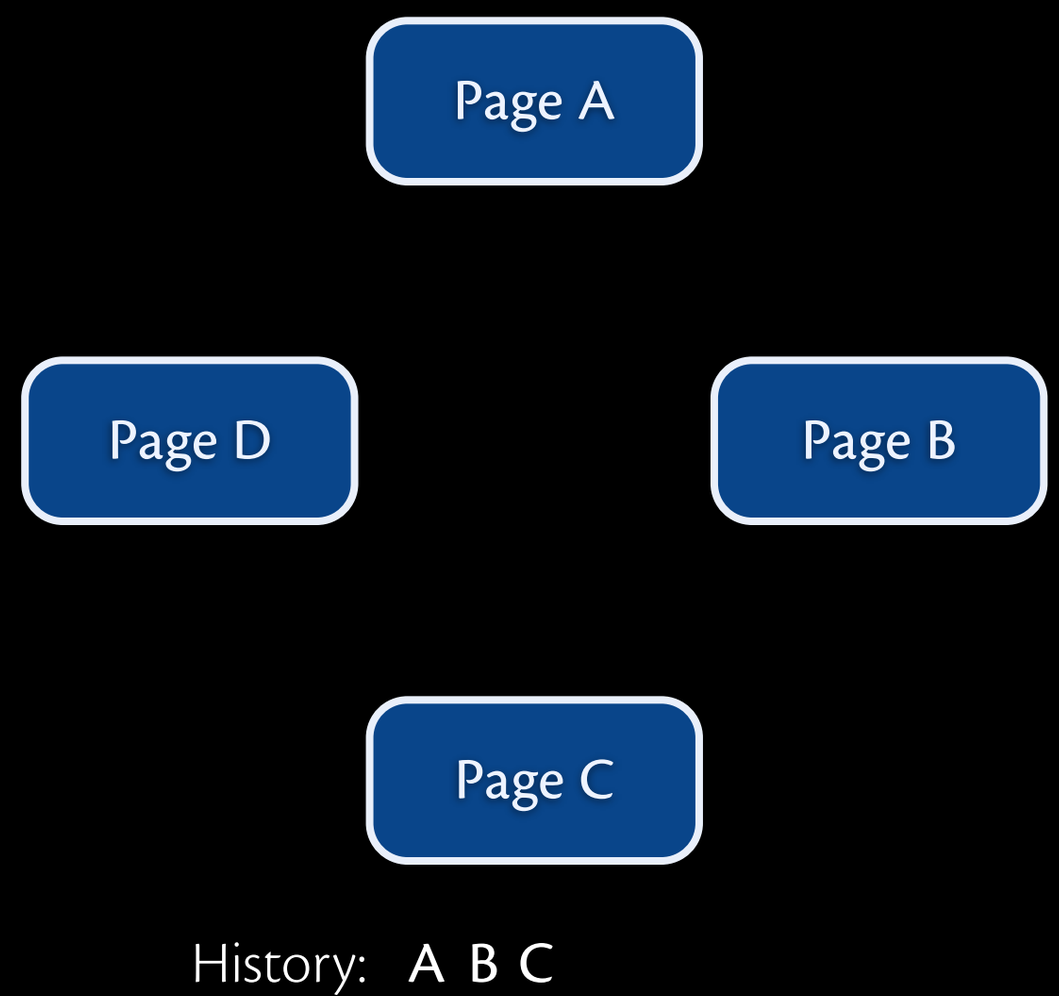
Cycling Detection

- Goal: recognize when users need help
- If user switches between viewing pages such that…
  - one page has been viewed at least three times and
  - another page has been viewed at least twice
- …then the user is probably looking for something and needs help
- Especially the case without browser "back" and "forward" buttons



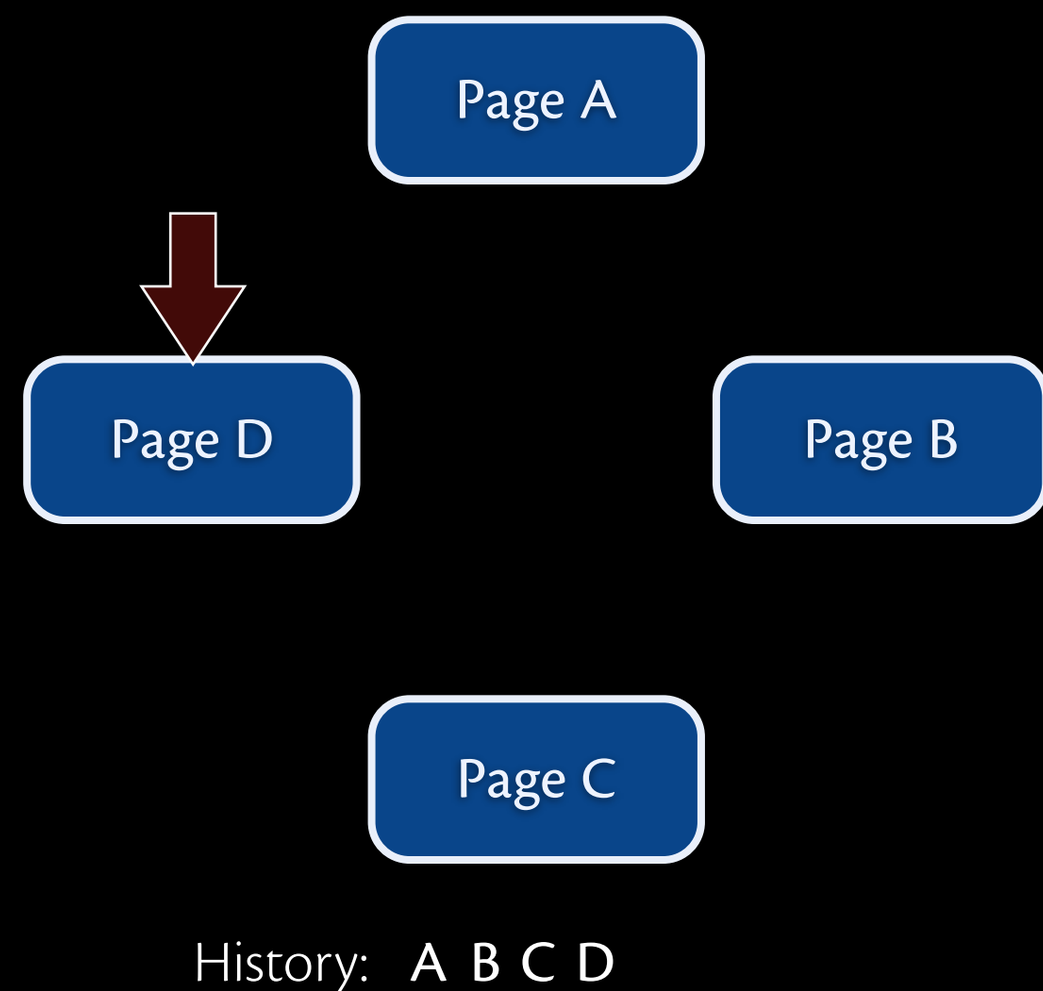
Cycling Detection

- Goal: recognize when users need help
- If user switches between viewing pages such that…
  - one page has been viewed at least three times and
  - another page has been viewed at least twice
- …then the user is probably looking for something and needs help
- Especially the case without browser "back" and "forward" buttons



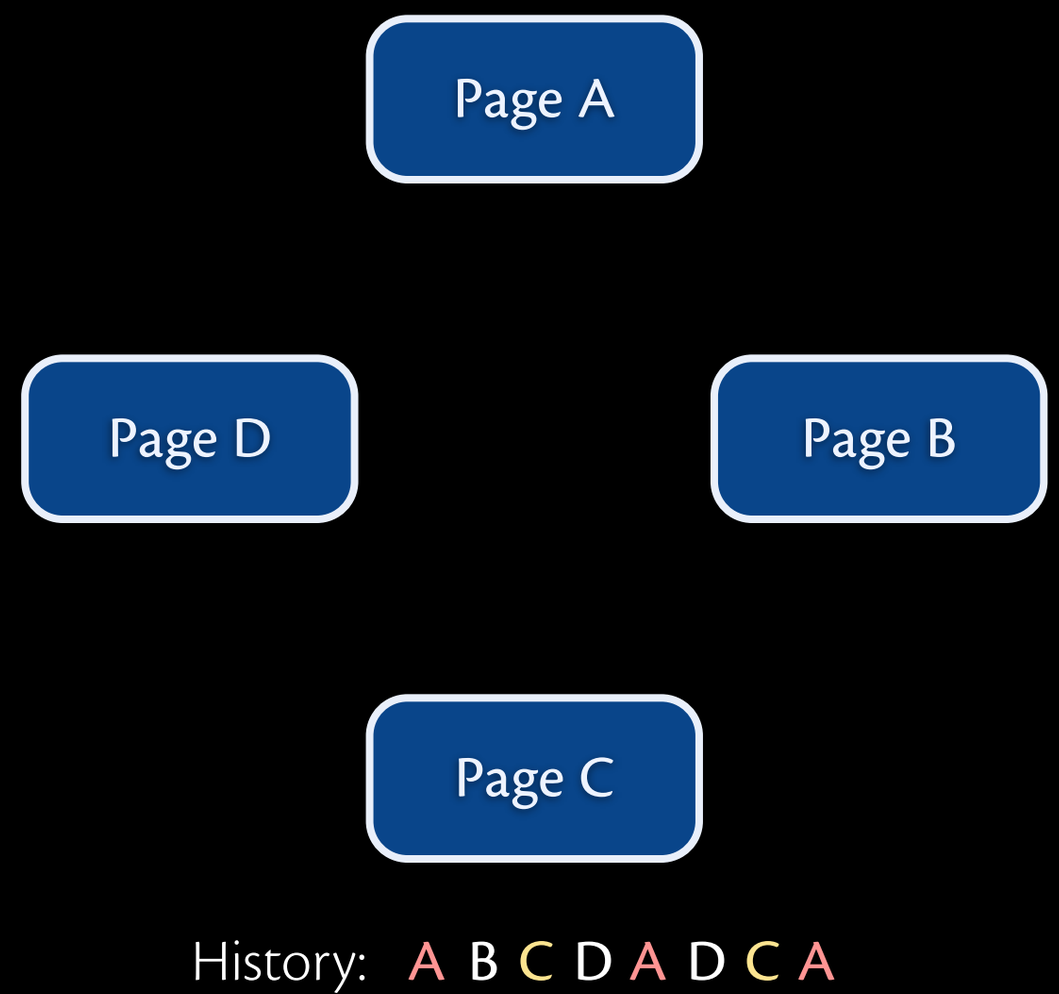
Cycling Detection

- Goal: recognize when users need help
- If user switches between viewing pages such that…
  - one page has been viewed at least three times and
  - another page has been viewed at least twice
- …then the user is probably looking for something and needs help
- Especially the case without browser "back" and "forward" buttons



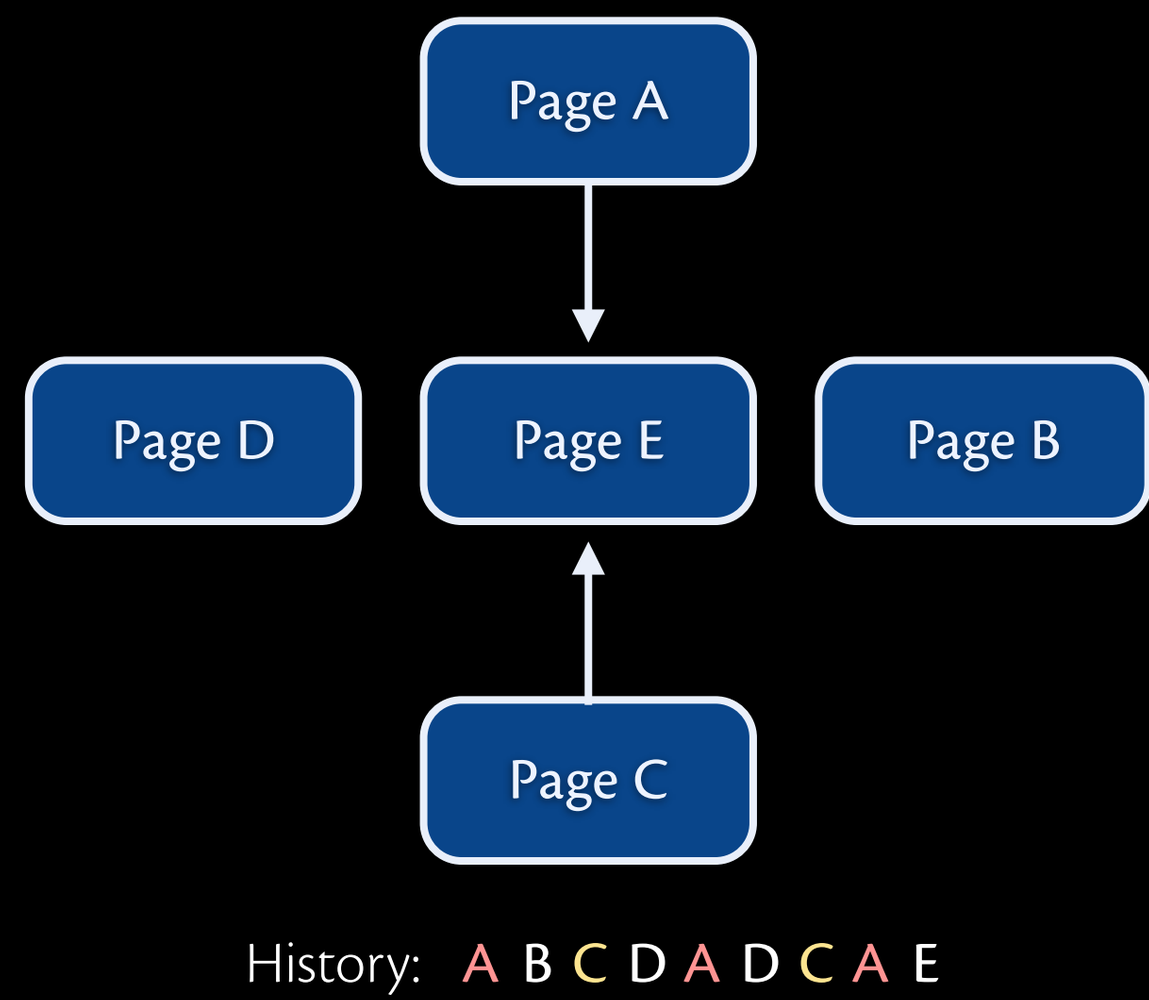
Cycling Detection

- Goal: recognize when users need help
- If user switches between viewing pages such that…
  - one page has been viewed at least three times and
  - another page has been viewed at least twice
- …then the user is probably looking for something and needs help
- Especially the case without browser "back" and "forward" buttons



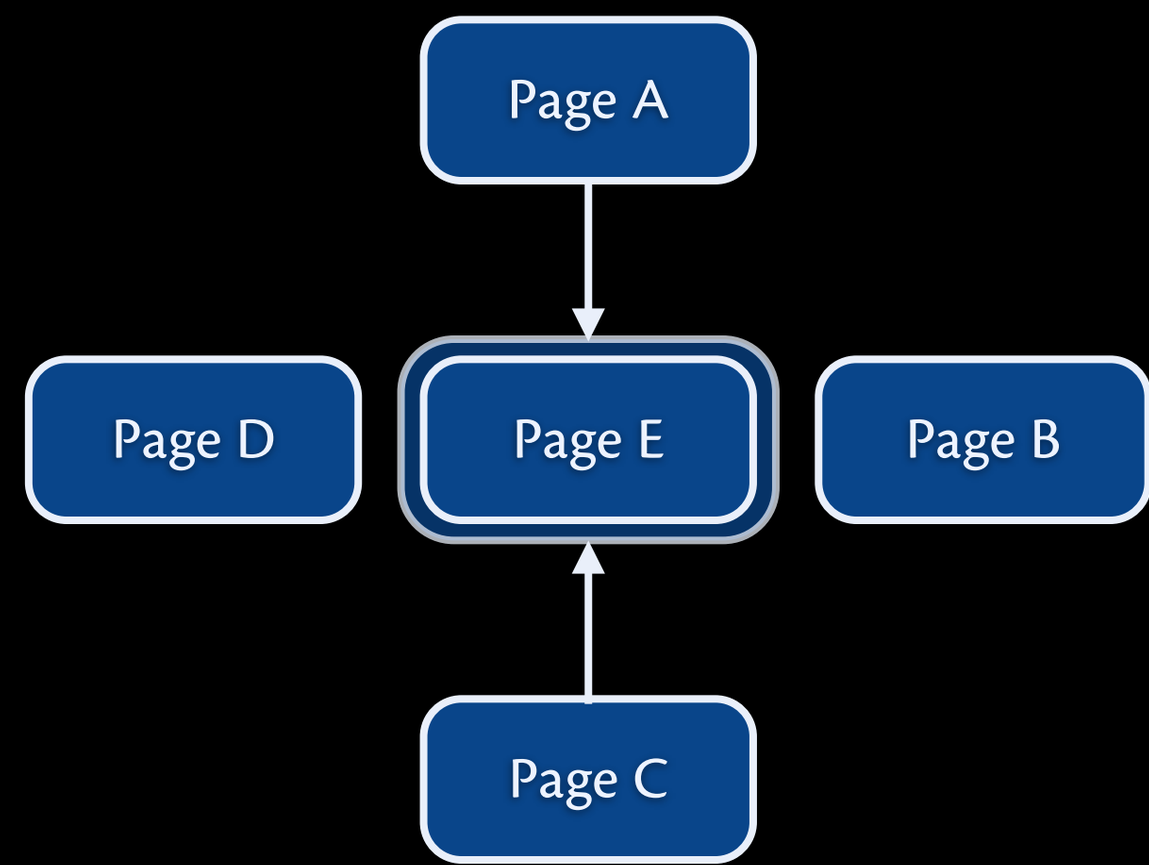
Cycling Detection

- When user eventually solves his or her problem, extra smart links are created
- Upon detection of cycling, system displays the extra smart links created by past users
- New smart links are stored in a weighted queue to evict old smart links not used



Cycling Detection

- When user eventually solves his or her problem, extra smart links are created
- Upon detection of cycling, system displays the extra smart links created by past users
- New smart links are stored in a weighted queue to evict old smart links not used



History: A B C D A D C A E

Featured Removed

- Annotations
  - Scheduled posting of audio files to central server
- Smart links
  - Distributing to other users
- Expert contact information

Questions?