

Visually UXposed

JUNE 10TH, 2016 – FINAL PRESENTATION

HAKIM & IVAN

The Domain Expert

- ▶ uxprobe is a tech start up developing expertise in user experience measurement
 - ▶ Targets IT managers for application improvement
 - ▶ Tracks how users interact with the software in real time
 - ▶ Measures success and satisfaction of users in context
- ▶ Paul Davies, co-founder of uxprobe was our domain expert



The Goal and Tasks

- ▶ Goal : Support the IT project manager in deciding what changes to make to an application
 - ▶ Changes such as: adding or removing features, changing structure, etc
- ▶ Tasks : the IT manager evaluates user experience information,
 - ▶ investigates the way the user interacts with the application, his success and satisfaction
 - ▶ uses the visualisation tools provided by uxprobe
- ▶ Our objective : design a novel visualisation tool

The Dataset

- ▶ Dataset: session logs of truck drivers using a fleet management application
- ▶ The structure of the data is
 - ▶ User ID (missing for 50% of the data)
 - ▶ Session ID
 - ▶ Task Name (ex: New Message Notification)
 - ▶ Event Name (ex: Notification: Open Message)
 - ▶ Event Type (screen, feature, error)
 - ▶ Time Stamp (! up to seconds)
- ▶ Volume: 1.2 million records, 161 MB
- ▶ Entities have exponentially decreasing frequencies

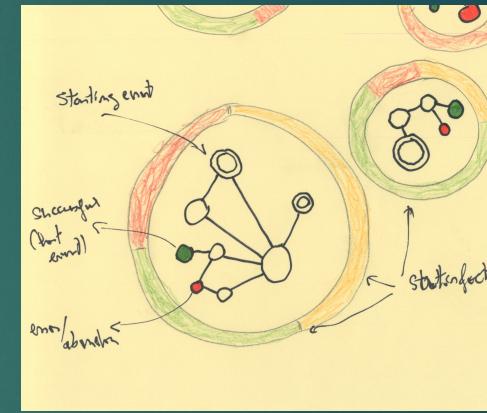
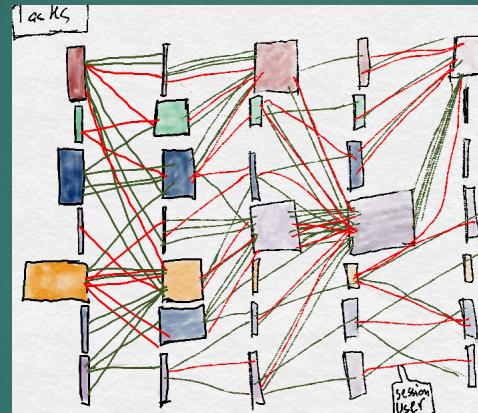
The Project

- ▶ Initial brainstroms before dataset
- ▶ Obtain the dataset, first developments, explore
- ▶ Meet with Expert for explanation, goals and tasks
- ▶ Several iterations of brainstorms and designs
- ▶ Several iterations of development, prototyping with hopes and setbacks
- ▶ Validation with expert, final decision on design
- ▶ Final development
- ▶ Blogs, project updates, report, final presentation
- ▶ 150 hours of development, 1300 lines of code

The Designs



The Designs: 3 candidates



The Designs

- ▶ Spheric representation of tasks, with graphs for events.
 - ▶ Issues: feasibility (conflicting sizes, visibility), very complex implementation, missing data
- ▶ A flow perspective, representing paths through the system
 - ▶ Issues: time perspective not so important for domain expert, conflicting tasks and events perspectives, sensitive to data issues, little satisfaction and error data, difficult to evaluate before a fairly complex implementation
- ▶ !!! A Sankey flow already exists at UXprobe, so we didn't consider this
- ▶ A network perspective, vizualized in form of a matrix, combining task and event levels
 - ▶ Plus: robustness to data issues and missing data, handles well the task/event hierarchy, manageable implementation, etc
 - ▶ Our favored design
 - ▶ Usefulness and choice confirmed with the domain expert

The Matrix



The Matrix

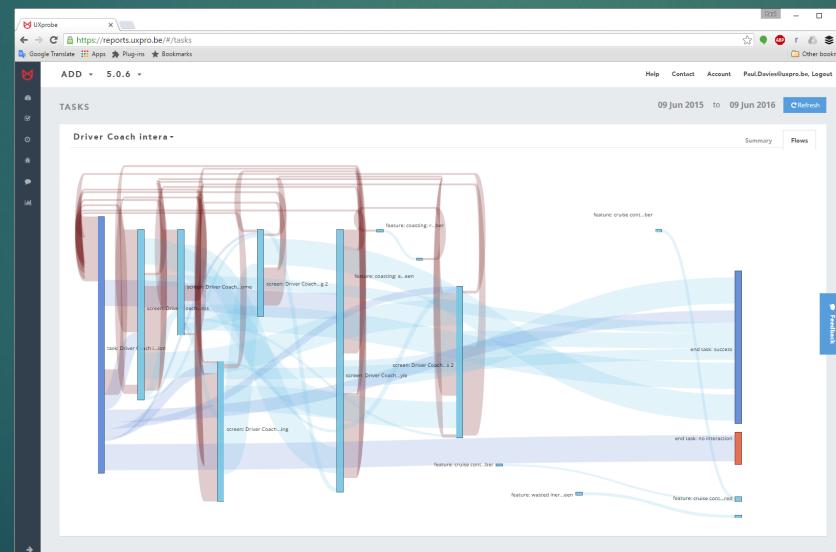
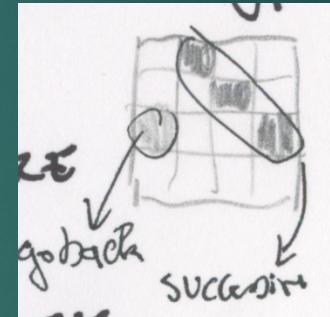
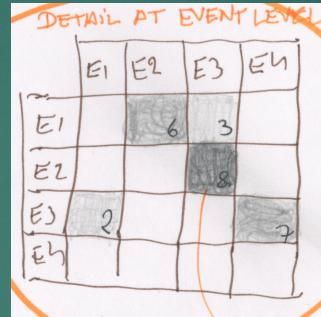
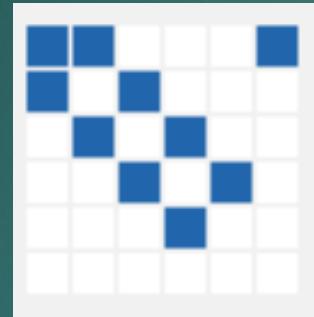
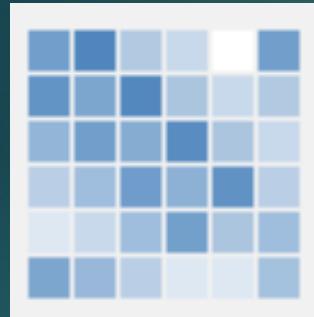


Default

Contrasted

Filtered

Design Highlights



The Matrices (lots of...)



Back To The Goal

- ▶ The matrix summarizes the entire application, yet allow for detailed analysis of patterns of use
- ▶ Complexity and structure of use across and within tasks is immediately perceived
- ▶ Some patterns of use are remarkable: “pillars”, empty lines, squares, sequences, isolation,...
- ▶ Errors are highlighted, their frequency, location and possible causes are assessed visually
- ▶ Miniaturisation still gives overall perception of use and can be used to contrast different user clusters or application versions

The Future

- ▶ Productive implementation???
- ▶ Improvements
 - ▶ Collapsible tasks
 - ▶ Frequency histograms for the tasks (could be shown along the edges of the matrix)
 - ▶ A master/detail design
 - ▶ Visual aids to easily locate the labels, such as guidelines, highlighting or magnifying the labels corresponding to a cell
 - ▶ Different sorts of sorting.
 - ▶ Use of satisfaction results of the micro-surveys, either as a sorting criterion, or surrounding the task tile, in the fashion of our first design.
 - ▶ Implement in D3



Backup slides

Implementation: Structure

- ▶ Three stages:
 - ▶ A preprocessing stage: data loading, recoding, cleaning, volume reduction, building data model and relationships, building the network, etc.
 - ▶ A processing stage: preparing the data in the required structure for the visualisation (i.e. the matrix view of the network of tasks and events)
 - ▶ The visualisation stage: drawing the actual visual (the matrix) and interaction (sliders, tooltip)
- ▶ Technologies:
 - ▶ Swift 2.2 for the preprocessing and processing stages
 - ▶ JavaScript and p5.js library for the final stage

Implementation Issues

- ▶ Time stamp is in seconds:
 - ▶ as we rely on the time to determine the sequence of events, events are likely to be out of their real sequence, because they appear have the same time.
- ▶ Some events are system events:
 - ▶ examples: “harsh braking: green => red”, “cruise control: amber => amber”, etc. : they are not initiated by the user

