frontpage-wact.pdf

Data Analysis as a Service: an infrastructure for storing and analysing the Internet of Things

Martin Lehmann

8th May 2015

1 Rant

1.1 Description of the interactive system

Type: NUI, (probably) TUI, no VR/AR/MR

Form factor: designed only for iPad 1/2/Mini, but can easily write responsive CSS to support more types of devices (relevant for touch tables, smart boards, touch walls, smartphones). Hopefully keyboard is connected. Designed to be used when sitting down, although can be just opened (not interacted with) to grok the project's state on the move. Designed to be interacted with only when necessary and looked at when information is required

Direct manipulation techniques: always inline manipulation, no external prompts (except virtual keyboard on the iPad if no external keyboard is connected (which it hopefully is). Deletion option (with confirmation) on each issue.

1.2 Sketches

1.3 Development process

1.4 Evaluation of the system in general

1.5 10 meaningful screens

Haha, 10. Each card state, hint, everything

1.6   Five actionable events

Card deleted (realtime updates of the board to see what is going on?), but mostly "the time is X and we have Y tasks left to complete", or "we have way too many tasks in progress, please fix"

1.7   Two valuable outcomes

1.7.1   Short-term valuable outcome

Immediately see and update the state of the project

1.7.2   Long-term valuable outcome

Be able to keep track of projects over time (but not across projects), be able to keep a continuous flow of issues

1.8   Practical stuff

Link to application:

1.9   Knowledge

NUI in software integration (obviously integrates with a different system, and provides valuable information about the current state of the company's project(s) and thus integrates with the company as well. Very easy to set up (many companies already have iPads hanging around just get some more, or open a browser).

Web client can be used by all types of devices (phones, tablets, touch tables, "smartboards", touch walls)

Motion controllers, gaze trackers, and brain trackers may be less interesting to use here (although Atlassian has been able to use brain power to move issues between swimlanes at a hackathon or so they say.

No specific SDKs have been used, although several frameworks are used:

- AngularJS (front-end) with ng-sortable, bootstrap 3, custom CSS

- Node.js (back-end) with several components (Express web server, Mongoose ODM (could just as easily have used relations), body-parser)

- MongoDB document database (could just as easily have used relations for this simple use case)

- Potential: JWT (or just cookies): save data inside the token

Important theories:

- Type of interface: somewhere in between a GUI and a NUI (the system has a GUI which is "optimised" for the natural interface touch, built for tablets with an existing touch screen. CLI only used for development :)

- Behaviour models: predictive models

- Hick-Hyman Law: about reaction time when presented with a bunch of options, not really relevant as there are zero menus (for which the law is mostly being used)

- Keystroke-level model (KLM): Key stroking only relevant when adding or editing issues, pointing relevant because the finger moves, homing time minimised because of large, responsive controls, drawing minimised (and destinations hinted), mental operator, system response operator (hello Heroku)

- Motor behaviour models: descriptive models: (state 0: waiting for stimuli) -¿ ¡finger down on issue handle (entire issue)¿ -¿ ((state 1: dragging issue) -¿ ¡finger up above legal area¿ -¿ (state 2: releasing "dropping" issue) -¿ (state 0: waiting for stimuli)) OR (¡finger up from text¿ -¿ (state 4: editing text) -¿ ¡tap "save changes" control¿ -¿ (state 0: waiting for stimuli))

- Mental models: HCI Design: target system (duh), conceptual model (issue tracker with touch interface which supports only adding and moving/reordering issues), system image (what is this?), user's mental model of the system image (hopefully very close, but discuss this), designer's user model (hopefully very close, draw contrasts and use same metaphors/stuff as Trello, JIRA, Asana, ding.io, ServiceNow, Symfoni Notes, other issue trackers users have already used, and simple post-it note setups)

- NUI types: only touch (skin is unrelated to the domain, but gestures (e.g., with a kinect in a meeting room) is interesting, speech is unrelated unless the audience has no arms/ability to touch (disabled), gaze tracking is cool but unnecessary unless same as speech recognition, brain machine interface only relevant for heavily disabled users who are probably not in need of an issue tracker)

- Spatial 2D NUI guidelines: environment is optimised for touch through large elements (15mm in all directions, at least 5mm between elements on tiny tablet screen, all touch targets are equally sized), does **not** allow users to change the layout because it is not needed, only one screen -¿ simple to position controls consistently (but consistent across swimlanes), spatial relationships: higher up means higher rank/priority/importance (it is a hierarchy), todo, doing, done are obviously positioned in that order, naturally consistent, not a multi-user system (although it could have been with multi-touch recognition, which is not supported by ng-sortable but could be interesting, would not let more than one user change everyone's view, hard to indicate ownership, only uses flat/wide "navigation" for controls, no hierarchies (e.g., dropdowns), main view has only important controls and objects, but is not too sparse

- Inter-user task coupling (irrelevant because no two users will use the system at the same time but could implement this with sockets so that both users will always know what the current status is)

- Touch input: Fat fingers: no buttons (only large areas), large margins, everything can be dragged (no specific drag handles), may cause trouble with scrolling

- Could be very cool to look at iceberg target implementation in a swimlane application

- All objects on screen (except swimlanes) are draggable/sortable through land on (start drag) and lift off (release to new position) except when that means "edit text inline", but sliding onto a swimlane's issue list when dragging an issue causes a hint for the next free position, and sliding off the entire board when dragging an issue results in the issue not being moved

- Sources of errors: Fat Fingers, multiple capture states, accidental activation (arm brush doing nothing to prevent this as of yet), physical manipulation constraints, stolen capture, tabletop debris (it's an iPad can't really get around that one)

- Gamification: no users (and no points)...

- Not using a mouse

- Not using a stylus (although recognising a stylus and allowing in-place drawing of "attachments" for issues would be interesting to look into)

- NUI gestures: not really used (should give escape event)

- INRC: **Identity** (move issue or start changing name always does just that), **Negation** (move back, change back always undoes could look into gestures like slide to delete, slide back to restore item which is already a thing), **Reciprocal** (no undo button or anything like it, but could easily implement one), Commutative (changing name, then moving = moving, then changing name)

Ethical problems: none in this application (except for security, duh), but certainly applicable

Short literature review

Conclusion: Stuff has been done, stuff is lacking, more stuff needs to be done in stuff areas

2   Introduction