Quill: A Collaborative Design Assistant for Cross Platform Web Application User Interfaces

WWW2013, Rio de Janeiro, 2013

Vivian G. Motti, Université catholique de Louvain
Dave Raggett, World Wide Web Consortium

This was supported by funding from the European Commission's Seventh Framework Program under grant agreement number 258030 (FP7-ICT-2009-5).

How mobile-ready are corporate websites?

- "Only 20% FTSE 100 corporate sites have mobile optimised content"
  - Nicola Thompson, Head of standards services, Magus, July, 2012
    - http://slidesha.re/ZdyLjI
- UK companies not ready for mobile internet
  - Two-thirds of companies in the FTSE 100 have websites that are difficult to use on smartphones, a study shows.
  - Robert Cookson, Financial Times, 2 January 2013
    - http://on.ft.com/Zp4UZI
    - "FTSE 100 companies are not mobile-ready and are wasting millions of pounds on internet advertising by sending visitors to websites that do not work as users expect them to," said Jonathan Bass of Incentivated
    - Domino's Pizza, one of the pioneers of online fast-food sales in the UK, reported in September that purchases from mobile phones were growing at almost 50 per cent a year and accounted for nearly a fifth of its online sales.

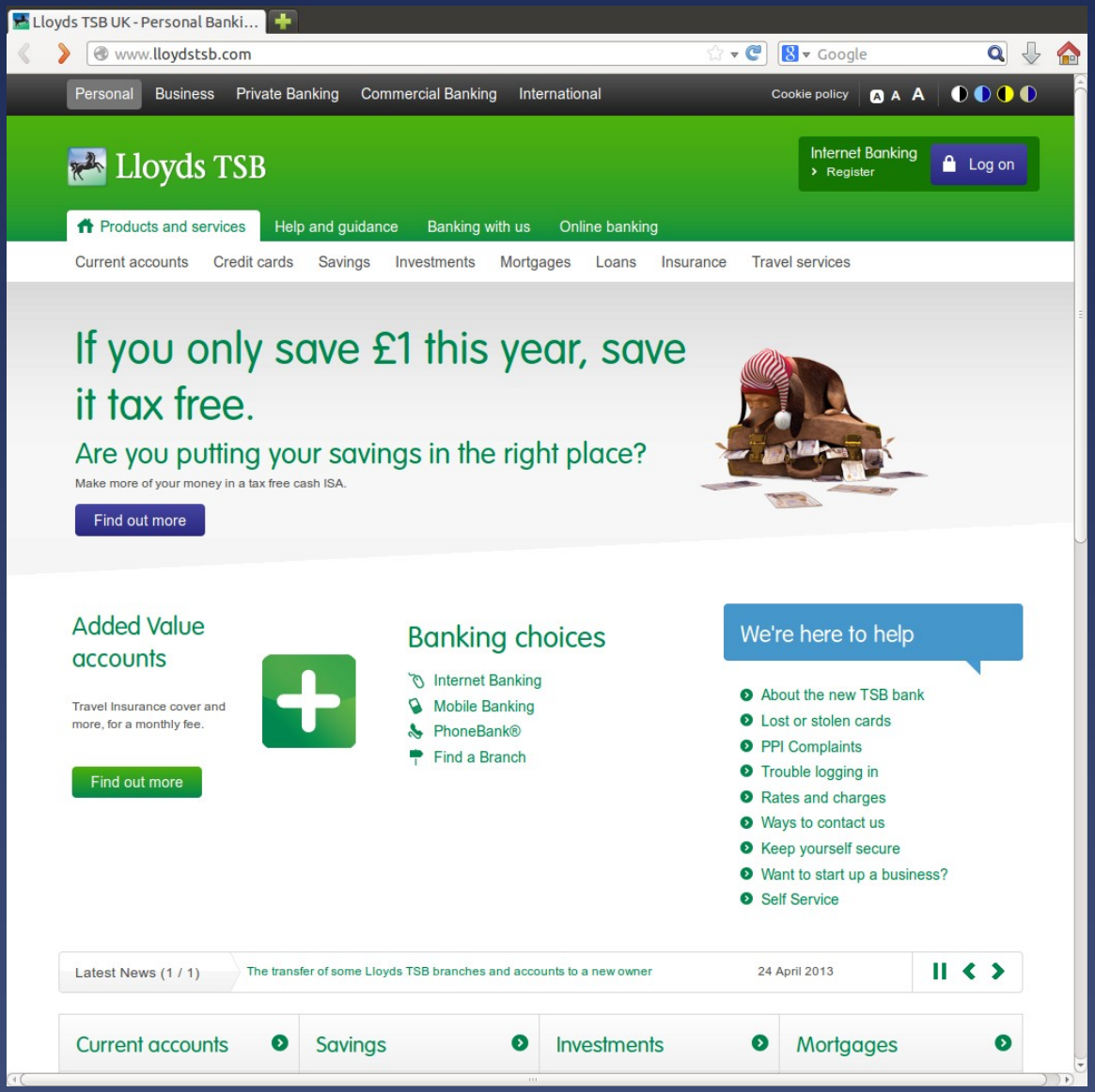
It's only going to get worse...

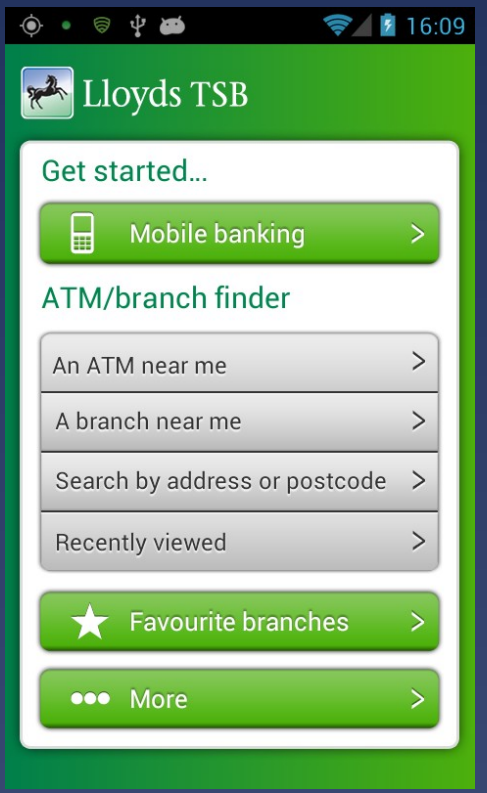
- Increasing variety of devices
  - Desktop, mobile, tablets, connected TVs
  - Regular and ultra high resolution displays
- Coming soon
  - Multi-screen applications
    - In the living room and in the office
  - Cars
    - Dashboard displays, smart phones, heads up displays, multimodal interaction, and concerns over ensuring driver safety
  - Wearable devices
    - Smart watches and glasses
    - Sports and healthcare

Why Web Apps?

- Current practice is to use web sites for desktop, and native apps for other platforms
- Developers need to learn new programming languages and SDKs for each platform
- Web technologies reduce the cost and increase the reach!
- Save time and money by avoiding app stores!
  - Keep all of your income
- Responsive Design techniques
- But need for better developer tools!

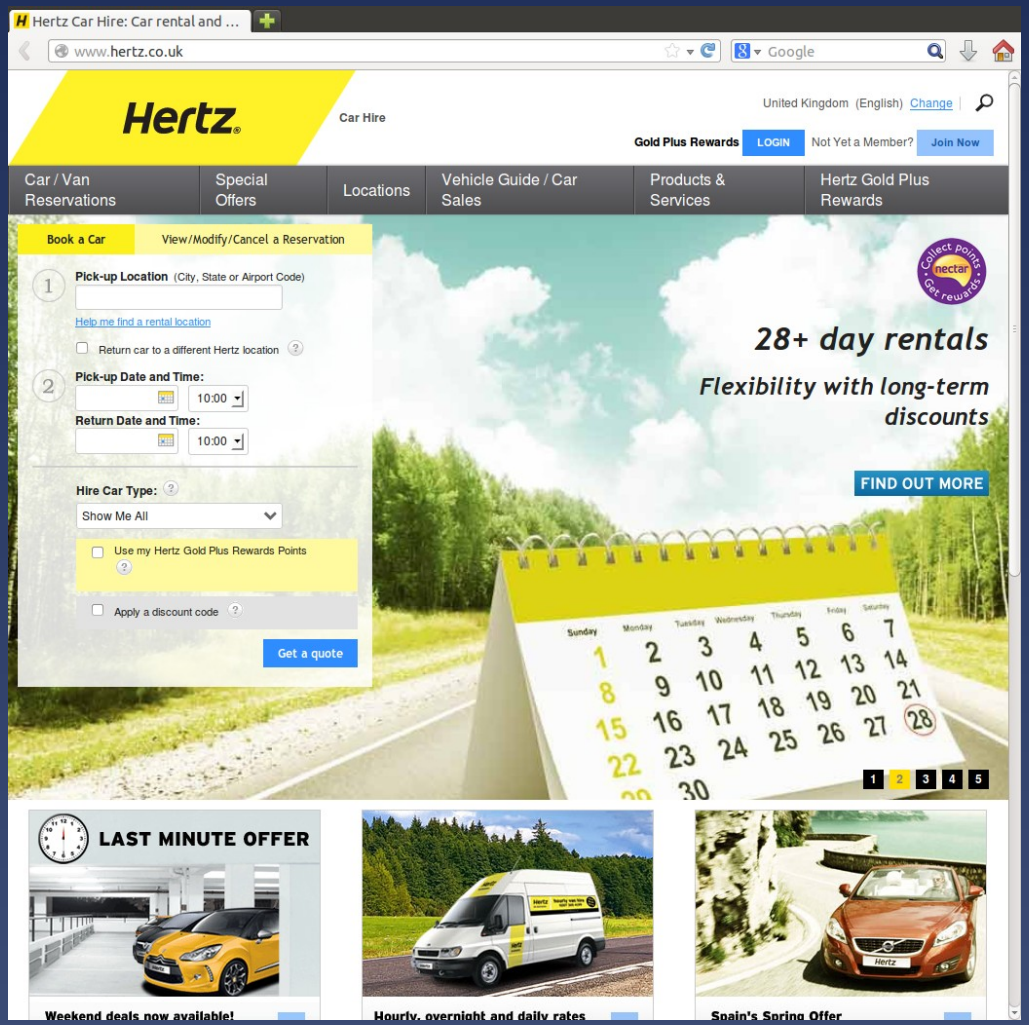
Desktop vs Phone

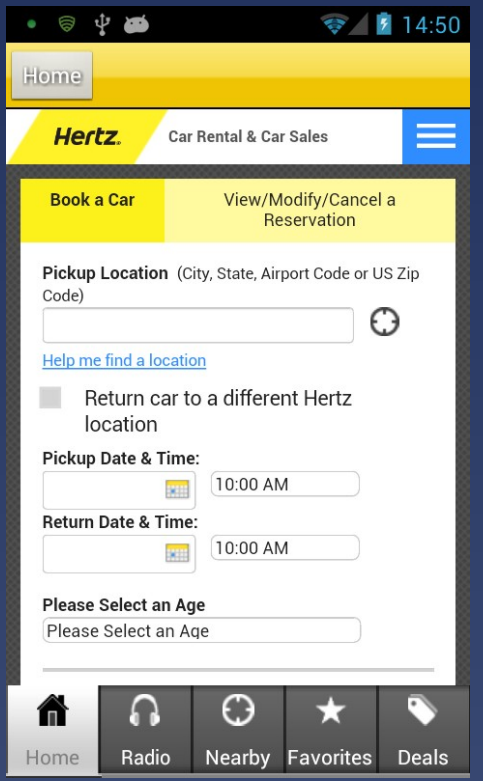




Captured from mobile App

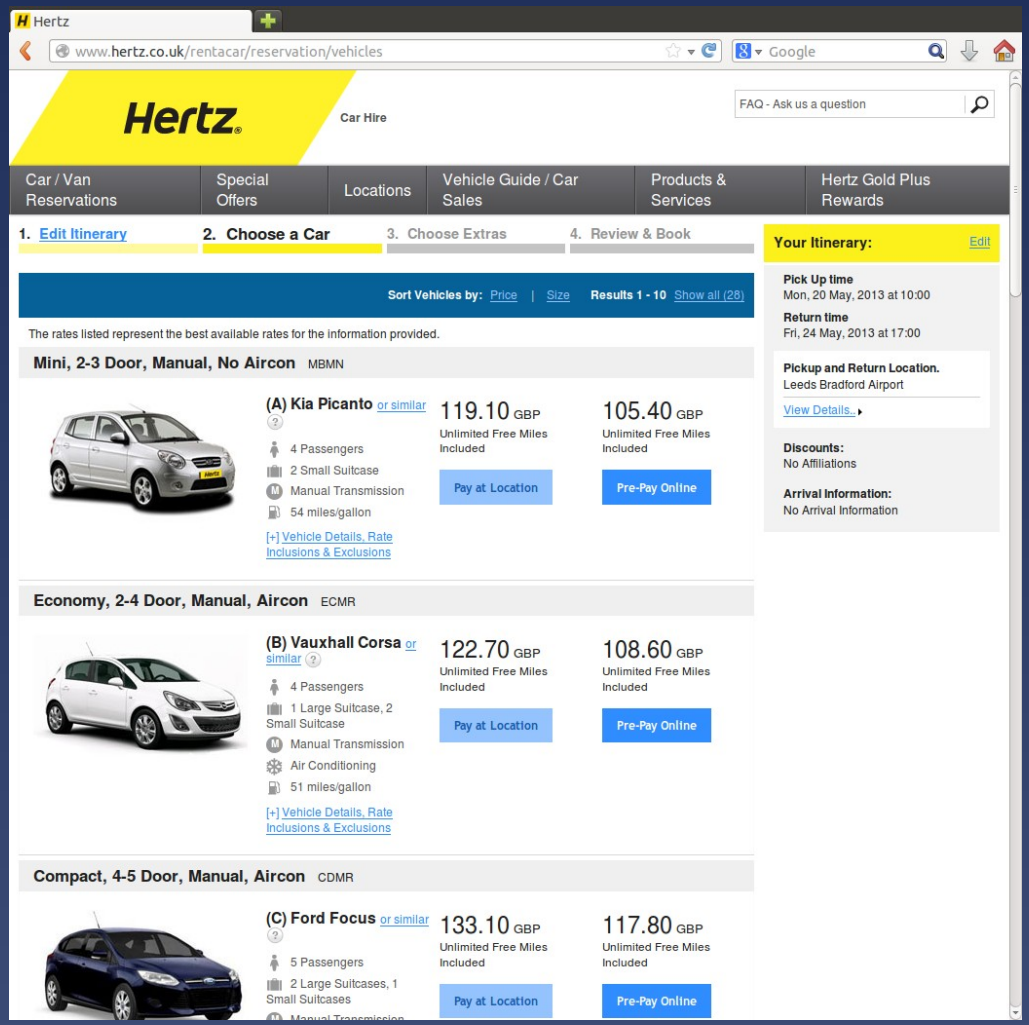
Desktop vs Phone

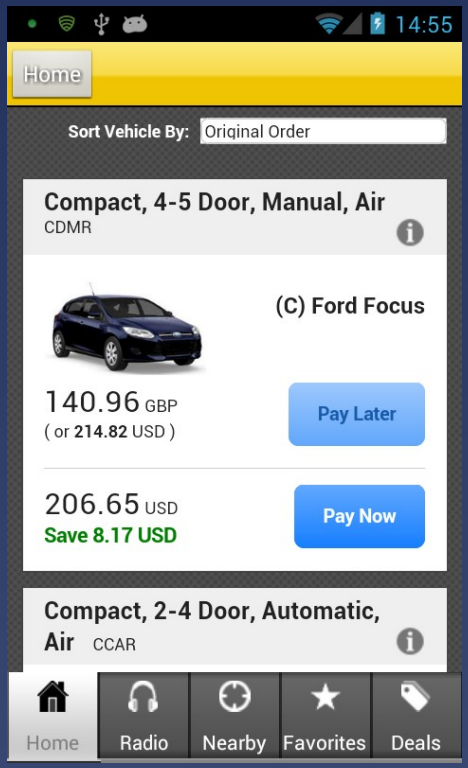




Captured from mobile App

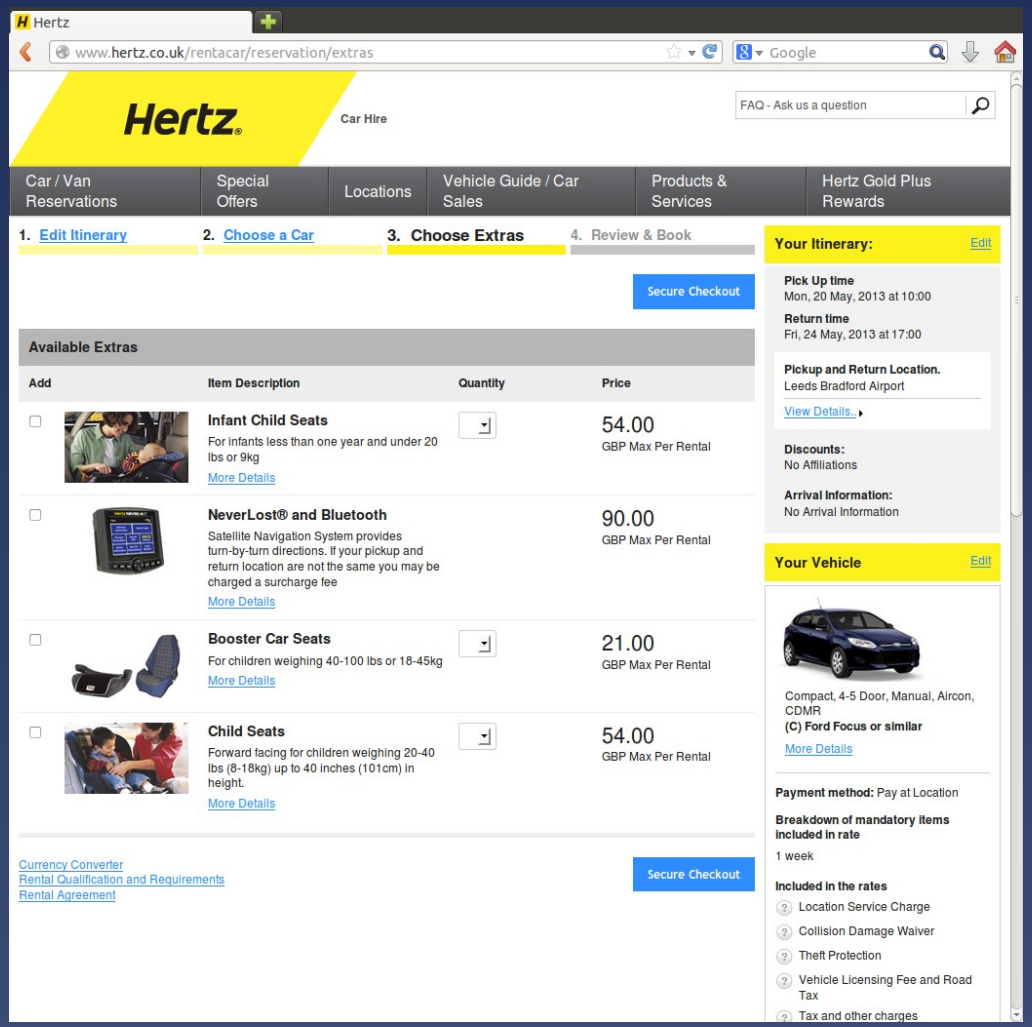
Desktop vs Phone

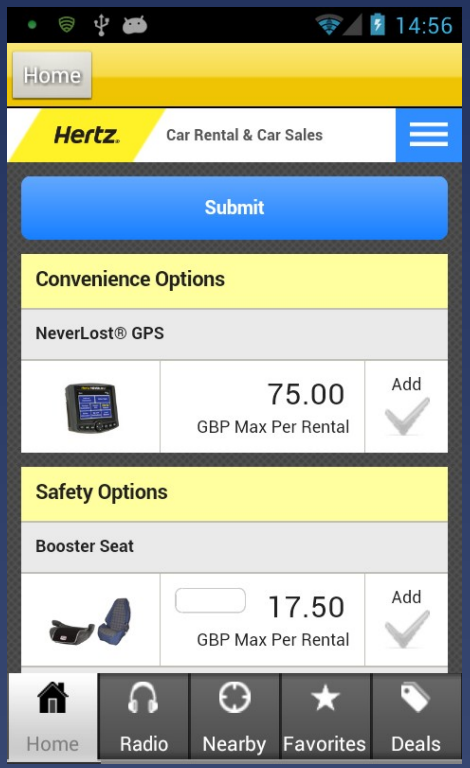




Captured from mobile App

Desktop vs Phone





Captured from mobile App

Design Process

- Agree on the business requirements
- Map them into domain and task models
- Use automated design tool to generate rough design proposals for each target platform
- Adjust the design to suit your taste
- Apply a UI skin and generate the final UI for each class of device
- Review and adapt until done!

Abstraction Layers

Domain

Tasks

Platform Independent

Abstract UI

Platform Dependent

Concrete UI

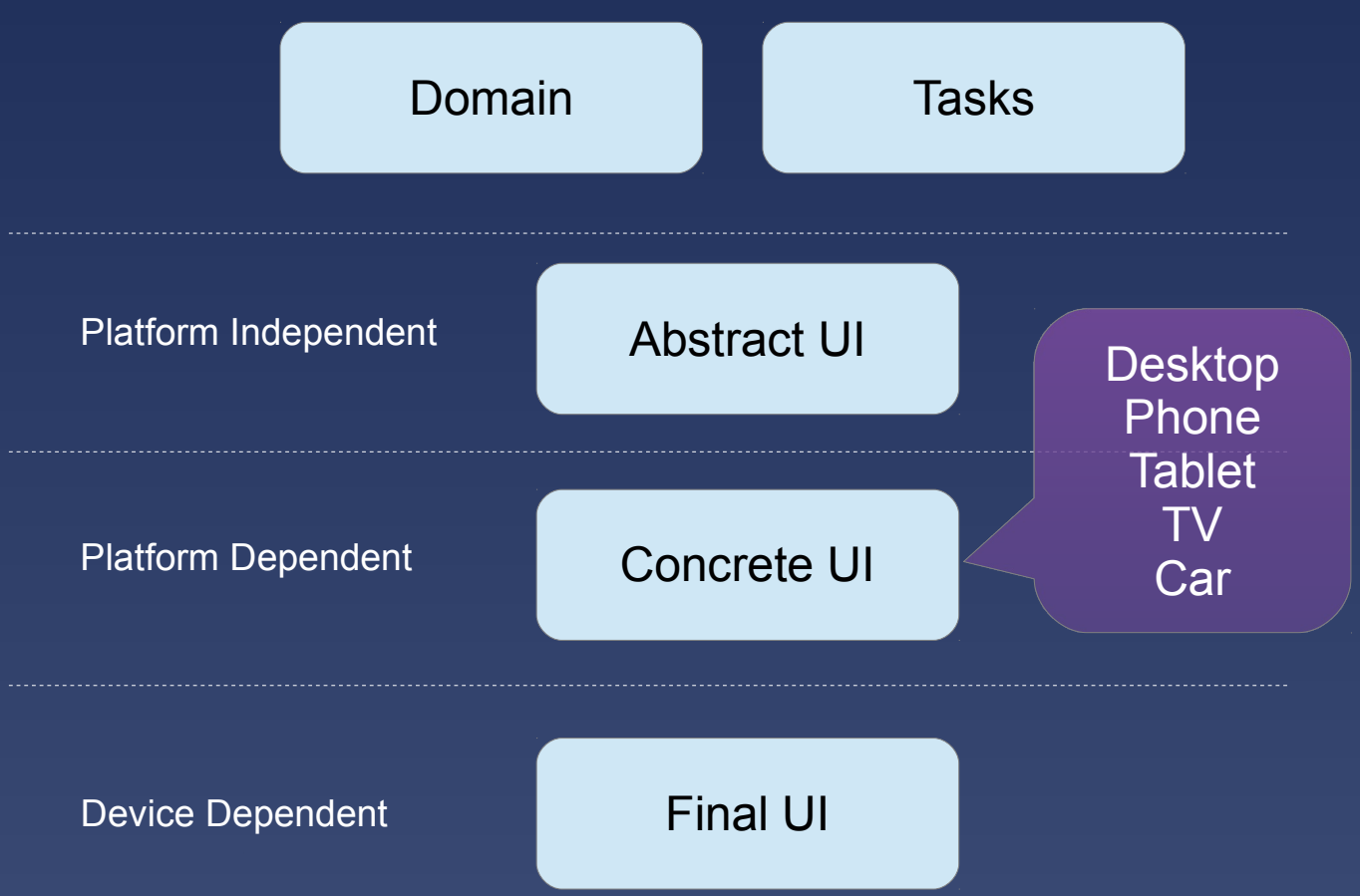
Device Dependent

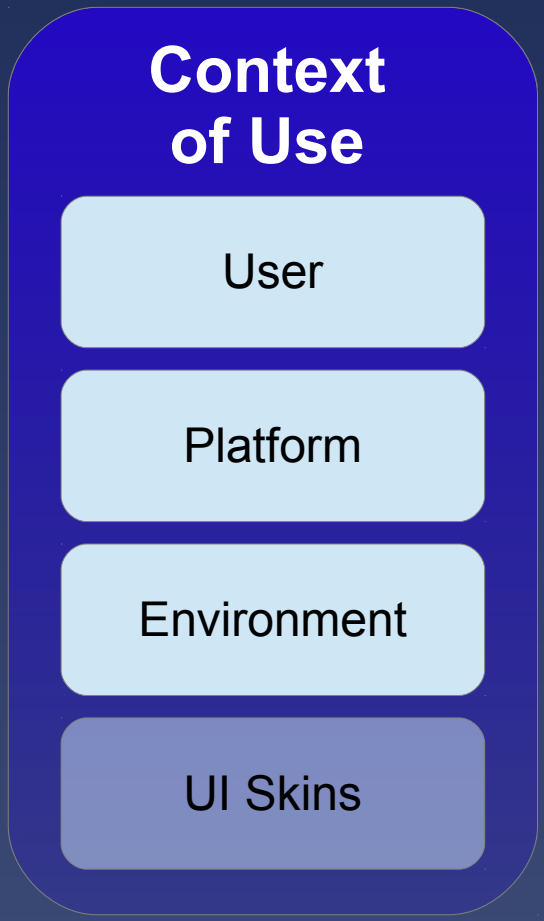
Final UI

Context of Use

- User
- Platform
- Environment
- UI Skins

Abstraction Layers





Abstraction Layers

Distinguish between normal and advanced features

Abstraction Layers

Domain

Tasks

Platform Independent

Abstract UI

Platform Dependent

Concrete UI

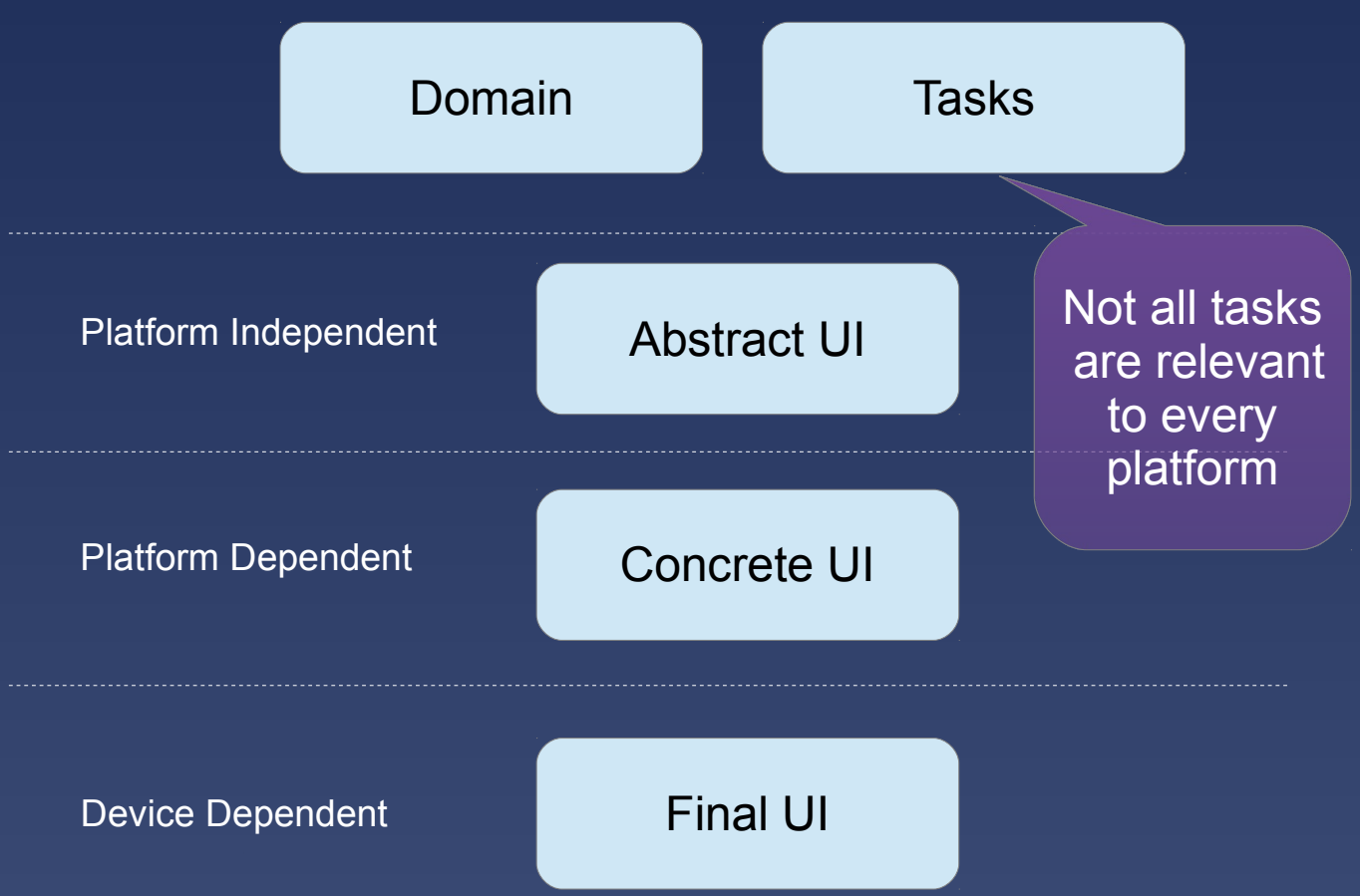
Device Dependent

Final UI

Context of Use

- User
- Platform
- Environment
- UI Skins

Abstraction Layers



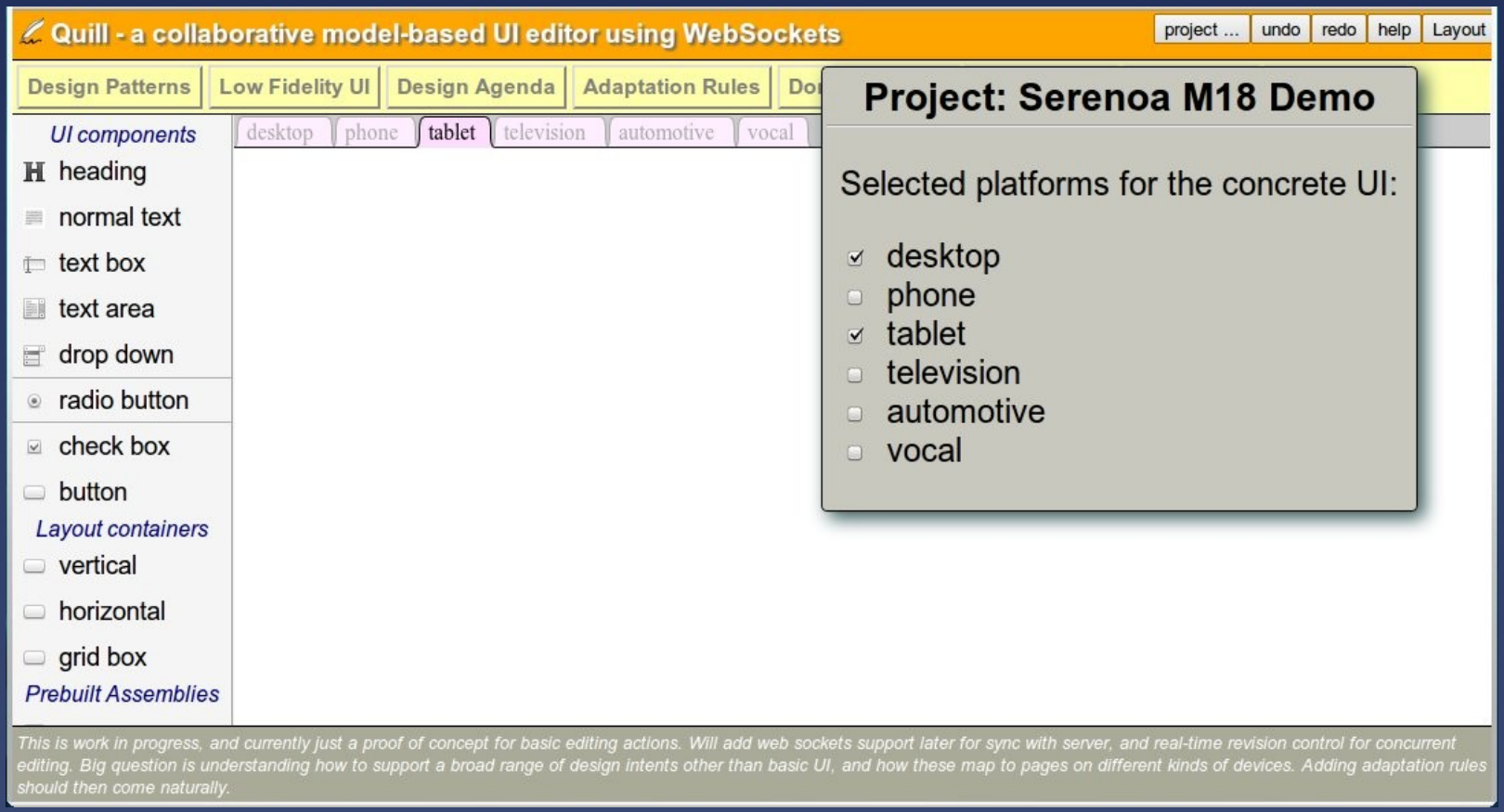
Context of Use

- User
- Platform
- Environment
- UI Skins

Quill

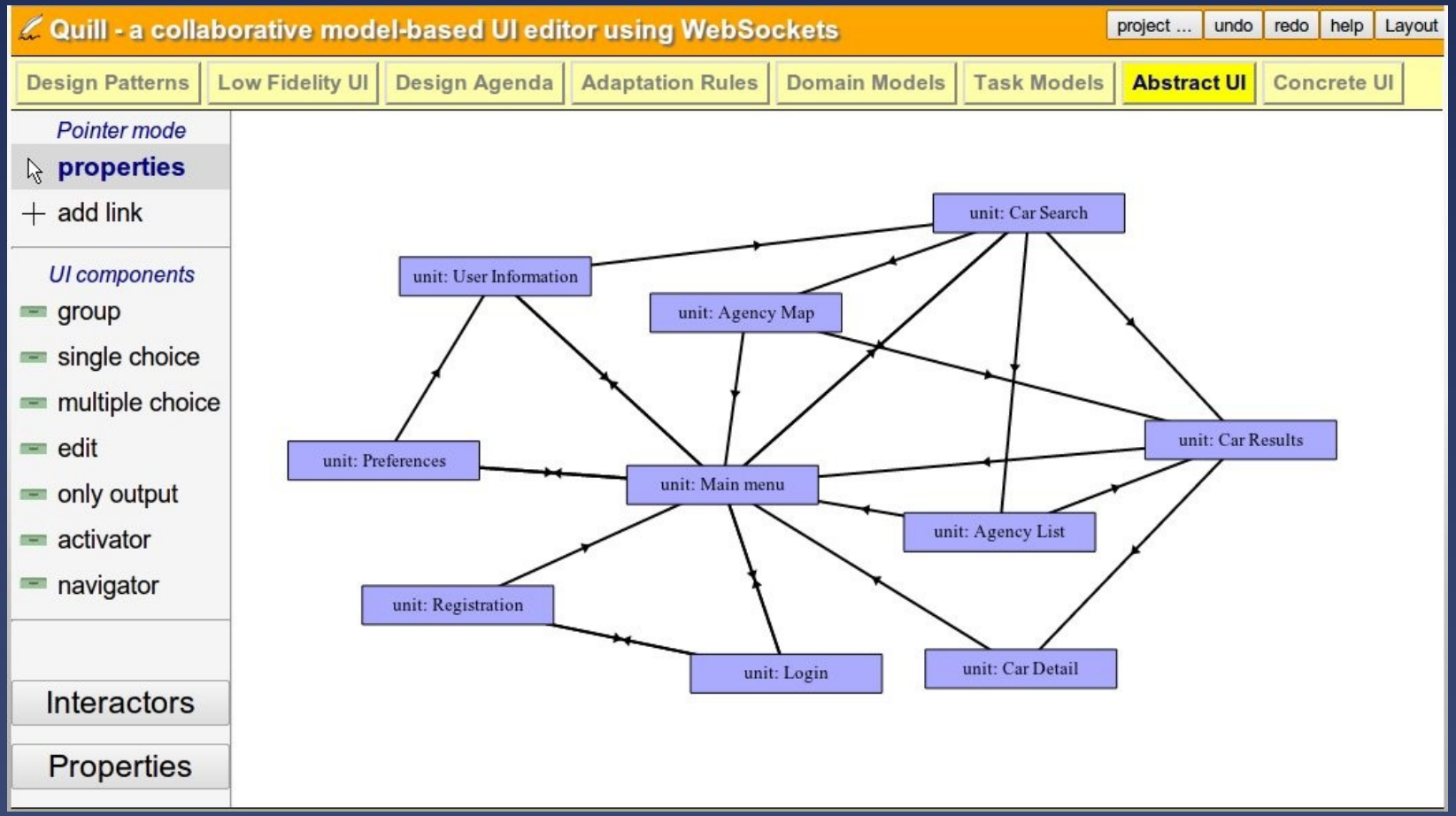
- Open Source, HTML5 based design tool
- Expert system generates proposals via a search of the design space
  - You select which one you prefer
- You can adjust the design at each layer of abstraction
- Your changes are considered as constraints and propagated to reduce the size of the search space as Quill looks for consistent designs
  - Quill synchronizes changes across layers and platforms
  - Ensures that the UI remains consistent across devices

Quill

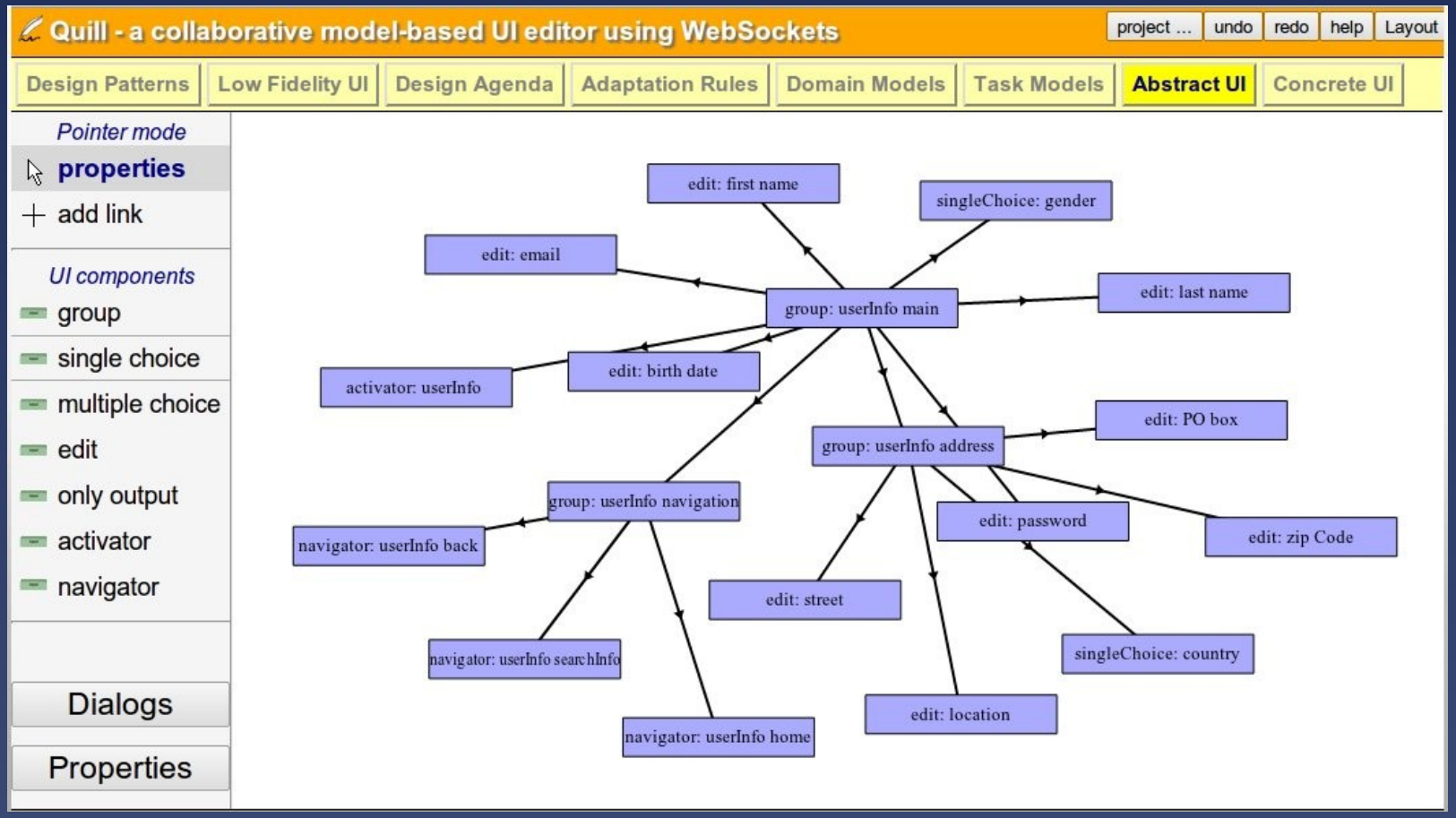


Early demo http://www.w3.org/2012/quill/

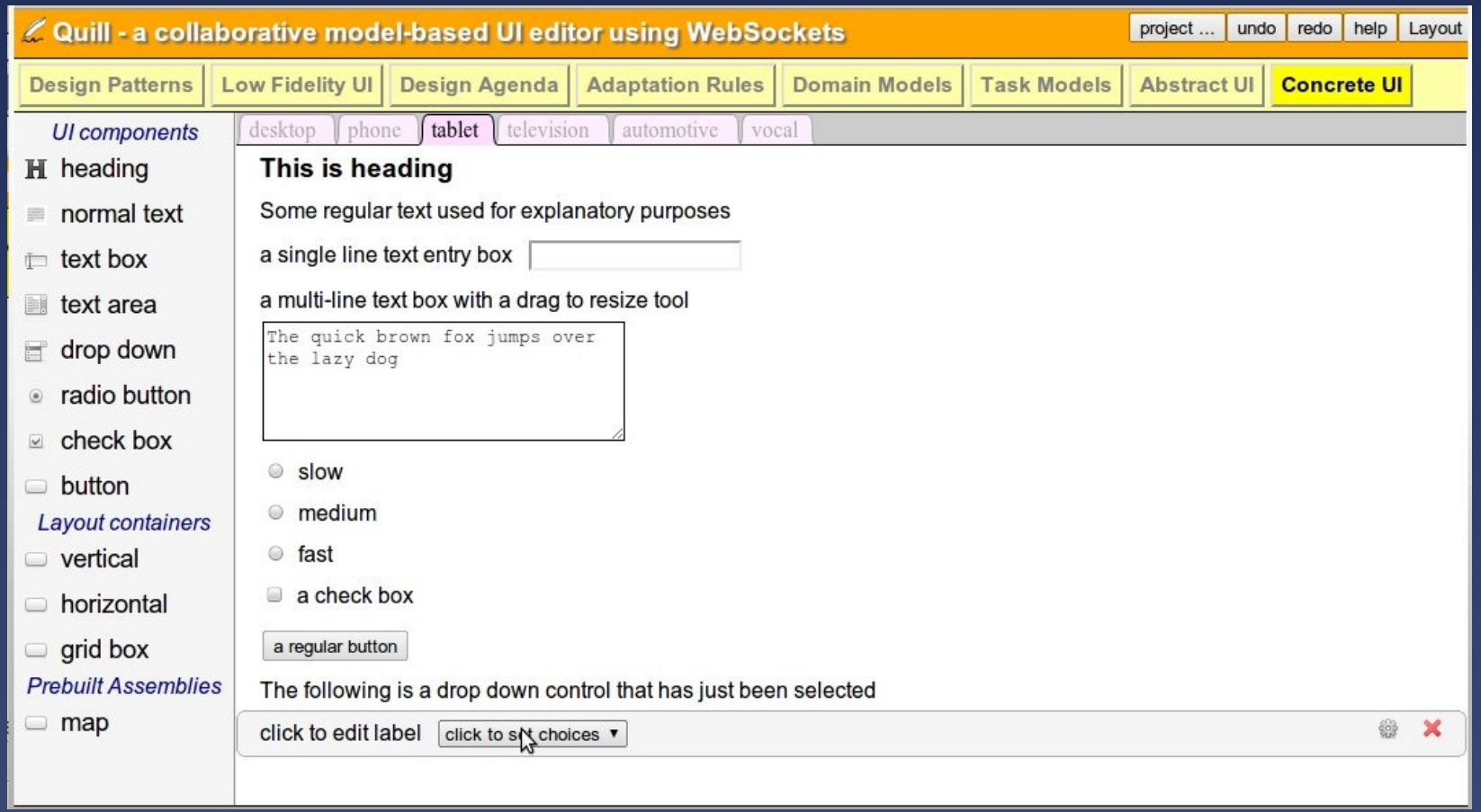
Quill – Abstract UI models



Quill – Abstract UI models



Quill – Concrete UI models



Domain Models

- Notation inspired by WebIDL
- Parsed into JavaScript Object model
- Annotated with
  - Relevancy conditions
  - Validity constraints
  - Default values
  - Call outs to handlers

```
@import "clauses"
@import "vehicles"

interface customer {
  first_name string;
  last_name string;
  email email_address;
  phone phone;
  young_driver boolean;
  marketing_opt_in boolean;
}

interface card {
  type {MasterCard, VISA, AMEX};
  number string;
  expiry_month month;
  expiry_year year;
}

interface itinerary {
  pickup_date date_time;
  return_date date_time;
  pickup_location location using find_location();
  return_to_origin boolean;
  return_location location using find_location();
  pay {now, at_location};
  #relevant return_location !return_to_origin
  #default return_location pickup_location
}
```

Task Models

- High level
  - Independent of UI details
- Hierarchy of tasks
  - Task – sub-tasks
  - Normal/advanced tasks
- Ordering constraints
  - One task enabling another
  - Unordered tasks
  - Task preconditions

```
task "make a reservation" {
  task "Customer details" using customer;
  enables {
    task "Pick up and return" {
      task "pick up details" {
        concurrent {
          task "find rental location" using
                 itinerary.pickup_location;
          task "enter date and time" using
                 itinerary.pickup_date;
          task "Return to pickup location" using
                itinerary.return_to_origin;
        }
      }
      task "return details" {
        concurrent {
          task "find location" using itinerary.return_location
            #precondition !itinerary.return_to_origin;
          task "enter date and time" using
                itinerary.return_date;
        }
      }
      task "Pay now or at pickup" using itinerary.pay;
    }
    task "pick car model" using agreement.vehicle;
    task "Choose extras" using extras;
    task "Review and book" using agreement;
  }
}
```

Abstract UI Models

- Platform independent
- Generated from domain and task models
  - Bindings to interfaces defined in domain model
  - Annotated to express constraints, e.g. relevancy

```
group "make a reservation" {
  group "Customer details" {
    select customer.first_name;
    select customer.last_name;
    select customer.email;
    select customer.phone;
    select customer.young_driver;
    select customer.marketing_opt_in;
  }
  group "Pick up and return" {
    group "pick up details" {
      select "find rental location"
            itinerary.pickup_location;
      select "enter date and time" itinerary.pickup_date;
      select "Return to pickup location"
            itinerary.return_to_origin;
    }
    group "return details" {
      select "find location" itinerary.return_location
        #precondition !itinerary.return_to_origin;
      select "enter date and time" itinerary.return_date;
    }
    select "Pay now or at pickup" itinerary.pay;
  }
  group "pick car model" {
    using agreement.vehicle;
  }
  group "Choose extras" {
    using extras;
  }
  group "Review and book" {
    using agreement;
  }
}
```

Quill's Architecture

- Models held in the cloud
  - Node.js based processor
- Direct manipulation interface in browser
  - HTML5 Canvas for graphical models
  - Force directed layout (charges and springs)
    - Uses window.requestAnimationFrame() for smooth animation
    - Plan to combine with hierarchical Voronoi cells for tree models
- Changes to models transmitted in text format over web sockets
  - Live editing through near real-time revision control
- A work in progress, with deadline of September 2013

Constraint Propagation

- Decision points – where human designer can force a given choice
  - e.g. decisions on layout and sub-dialogs
- Some things follow naturally from domain/task model
  - Changes to abstract or concrete UI that effect domain/task models
- Relationships across abstraction layers
  - Abductive Reasoning
    - If you know certain facts and also that certain relationships hold true, then it is possible to infer additional facts that must be true if the relationship is to hold
  - Formalized as combination of relational joins and unification
    - Demo http://www.w3.org/2013/01/abduction/
  - Replaces lots of event-condition-action rules

Developer Survey - Quotes

- Yes, I believe that models are very relevant and useful but the lack of "easy to use" applications, "easy to draw models" puts a certain level of resistance of using this tools from part of developers"
- ... if the model-based approach is directly responsible for the generated code and any changes in the code automatically reflects in the model then it would be extremely relevant to have this kind of approach in my development phase. The model would give me a macro approach to my application helping me to quickly understand what the application is doing...
- ... maintaining the docs and the code in a disjoint manner makes me waste some of the time [...] given that the coding sometimes needs to be changed to work
- I'm not sure models could be used in our domain: UIs are very complex and uses custom widgets.

Related Work

- EU FP7 Serenoa project
  - Context aware model-based user interfaces
  - http://www.serenoa-fp7.eu/
- W3C Model-Based UI Working Group
  - Standardizing task models and abstract UI
  - http://www.w3.org/2011/mbui/
- Responsive Design, e.g.
  - http://www.w3.org/2013/Talks/responsive-design.pdf
- This talk and associated short paper
  - http://www.w3.org/2013/Talks/quill-slides-www2013.pdf
  - http://www.w3.org/2013/Talks/quill-paper-www2013.pdf