

Practicum Software Engineering

Josh Jonte

Who am I?

- Graduated from IUPUI in 2003 with a BS in Media Arts & Science
- Worked in the software industry for 10 years, .Net/C# expert
- Most recently work at Mesh Systems as a software architect
- Why healthcare?
 - Trying to find a niche
 - Looked at several industries (financial, legal, etc)
 - Healthcare seems to be the most laggard in IT = opportunity

Interests

- Tablet form-factor and its impact on computing in healthcare
- Web applications replacing native applications
- Ruby on Rails
 - Learning Ruby
 - MVC design pattern

Rx Interact

- Drug to Drug Interaction detection tool
- Form Factor: ideally tablet (iPad, Android tablets) but works in any modern web browser (Chrome, FireFox 3.5+, IE9+)
- Data set: <http://www.drugbank.ca>
 - The database contains 6814 drug entries including 1437 FDA-approved small molecule drugs, 134 FDA-approved biotech (protein/peptide) drugs, 83 nutraceuticals and 5192 experimental drugs. Additionally, 4438 non-redundant protein (i.e. drug target/enzyme/transporter/carrier) sequences are linked to these drug entries. Each DrugCard entry contains more than 150 data fields with half of the information being devoted to drug/chemical data and the other half devoted to drug target or protein data.
 - DrugBank is supported by David Wishart, Departments of Computing Science & Biological Sciences, University of Alberta.

Rx Interact

- Designed to be used by clinician.
- Use severity and likelihood information to display to the clinician the risk to the patient.
- Description of interaction
- Display drug information

Technical Information

- Browser-based application
- Front-end
 - Uses HTML5 features
 - Canvas to render graph in real-time
 - CSS3 properties to enhance visuals
 - jQuery to manipulate the DOM
- Back-end
 - Ruby on Rails 3.1 RC4
 - SQLite
- Hosted on the Rackspace Cloud
 - 256 MB RAM, 10 GB Disk
 - \$0.015 / hour = \$10.80 / month

Test Cases and QA

- Test Cases
 - Able to search for a drug and the appropriate matches were displayed.
 - Drugs that are known to interact are displayed as interacting in the application.
 - Drug names are clickable and display information regarding the drug.
 - Drug interactions are clickable on the graph and in the list and display the appropriate information regarding the interaction.
- My wife was my QA person.

What I learned

- Good data is hard to come by
 - DrugBank information included drug interactions but not likelihood or severity information.
 - Reached out to several organizations:
 - http://www.datec.lavoisier.fr/gb/not_bdd.asp?bdd_id=163
 - <http://www.druginteractioninfo.org/>
 - http://www.medbase.fi/sfinx/whatis_finx.htm
 - Only got one reply: *"Thank you for your interest in the University of Washington's Metabolism and Transport Drug Interaction Database. **The University does not offer individual student academic (no fee) licenses to the Database.**"*
 - Ended up having to use a random number (0-1) for likelihood and severity information.

What I learned

- Even curated data has errors
 - Had to write an integrity check when inserting drug interactions due to duplicates
 - Symmetric relations not always there.
- Drug names not always what you think they should be
 - Example: 1-Methyl-3-Oxo-1,3-Dihydro-Benzo[C] Isothiazole-5-Sulfonic Acid Amide

What I learned (project management)

- Software Development Life Cycle
- Waterfall Methodology
 - How to write
 - business requirements document
 - functional specification
 - technical design document
- Put together a project plan with
 - Milestones
 - Statuses
- Using Visio to draw application flow

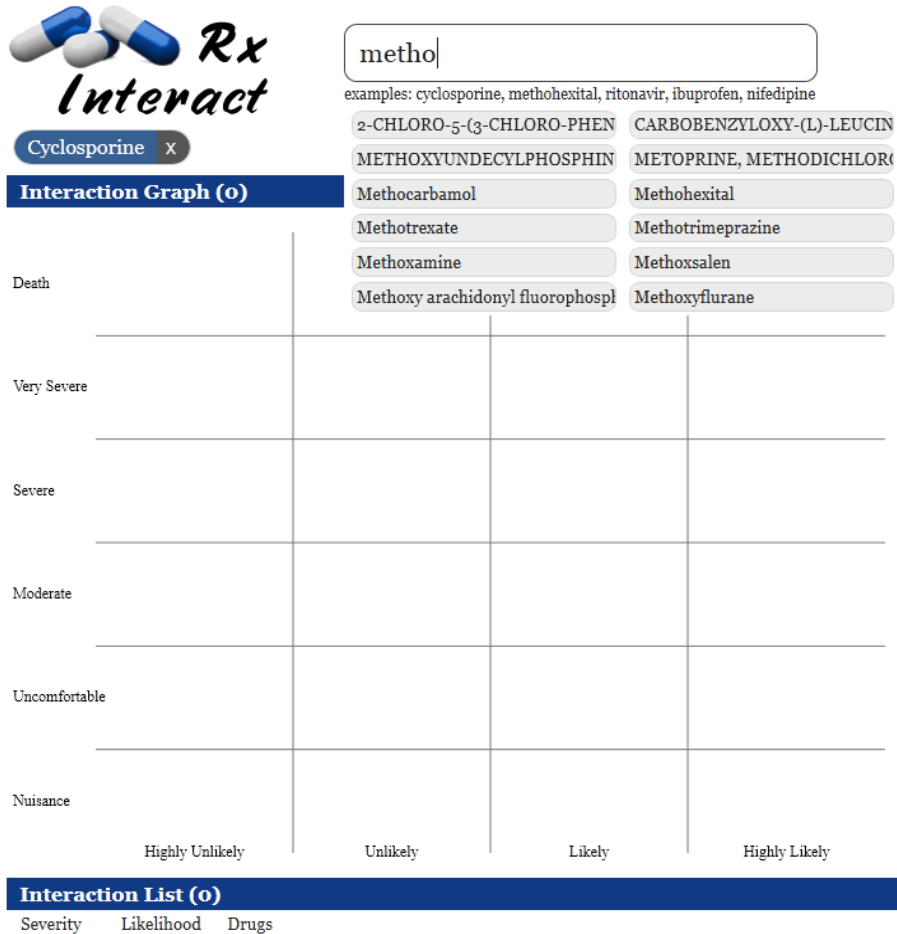
What I learned (technicals, front-end)

- 768 pixel-width is not a lot to work with (iPad in portrait)
- Interactions that work well on a computer do not translate to a tablet
- Rendering anything complex on a Canvas is tedious.
 - You get basic shapes like lines and arcs.
- jQuery is not fast on mobile devices.
- Creating web applications that look and behave native applications on iOS is pretty easy.

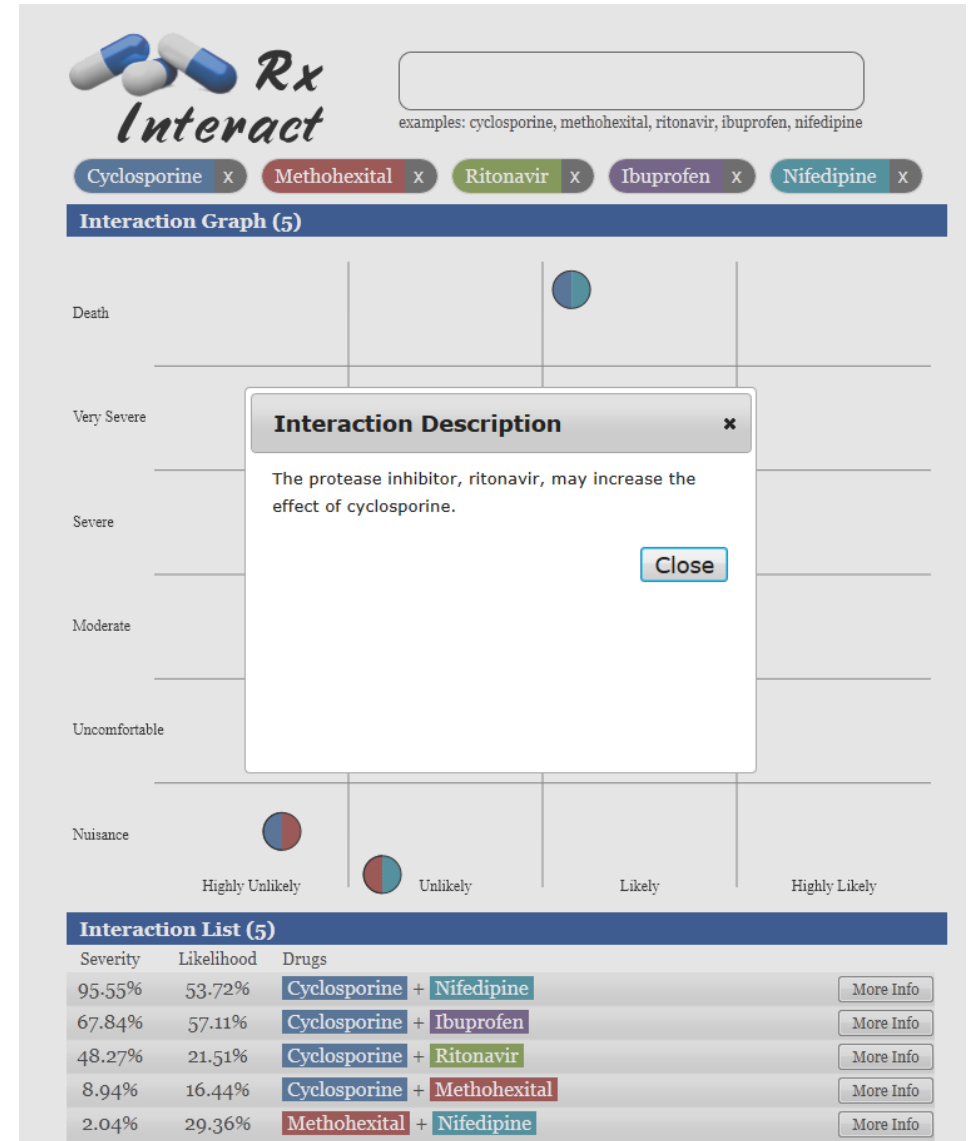
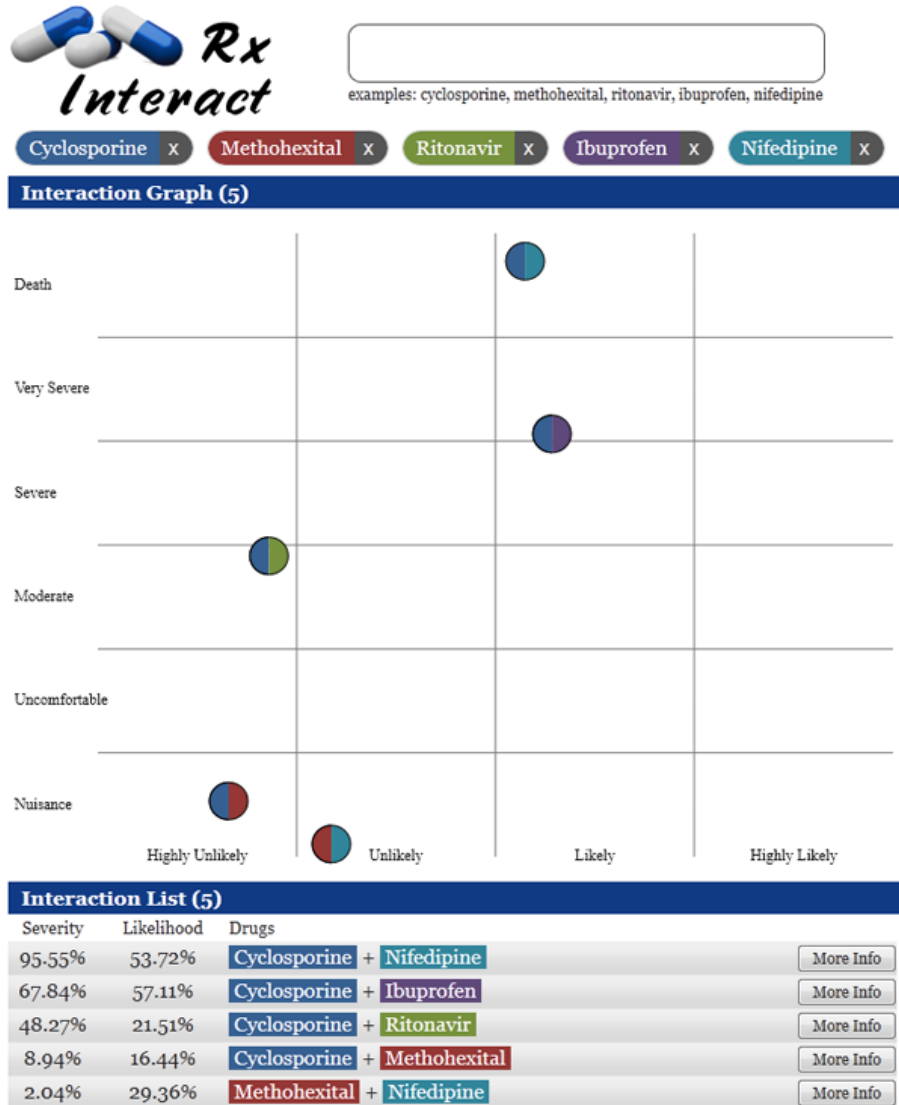
What I learned (technicals, back-end)

- I like Ruby 1.9.2
 - Terse yet readable
- Rails 3.1 is a very productive framework
 - Lets the developer focus on the business problem and not □ the web integration
- Ruby isn't very memory efficient.
 - 88 MB XML document = 1 GB of RAM used
- SQLite isn't very fast
 - ~30 minutes for importing + integrity checks
 - 6,826 drugs
 - 21,046 drug interactions
- Rails' built-in web server (Webrick) is incredibly slow
 - Ended up using a 3rd party web server, Unicorn

Screen shots



Screen shots



Demo

<http://50.57.110.175:8080/>