A Modelling Language for Interactive Web Applications

ASE 2009 Doctoral Symposium

13 November 2009

Jevon Wright j.m.wright@massey.ac.nz

Dr Jens Dietrich j.b.dietrich@massey.ac.nz

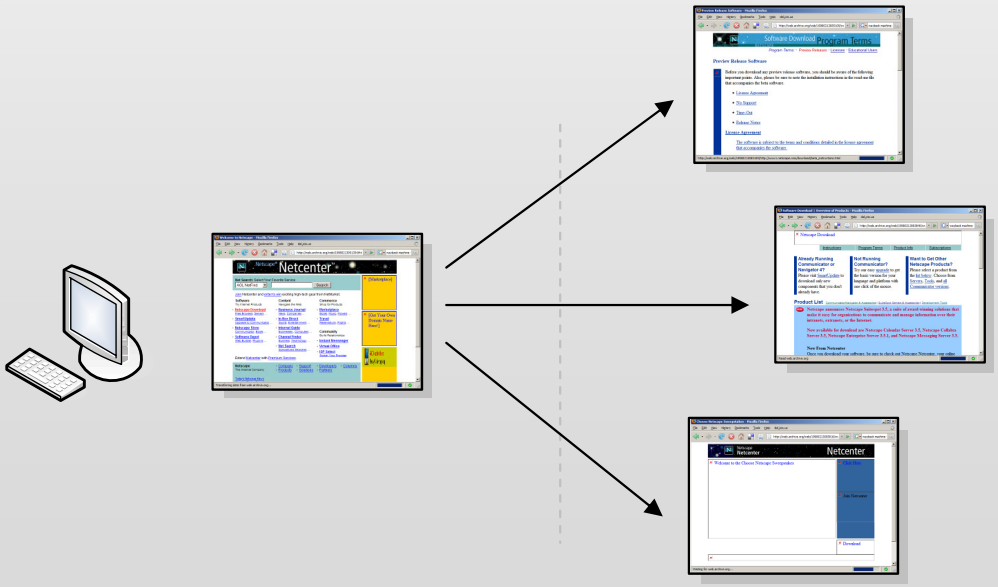
Massey University
Palmerston North
New Zealand

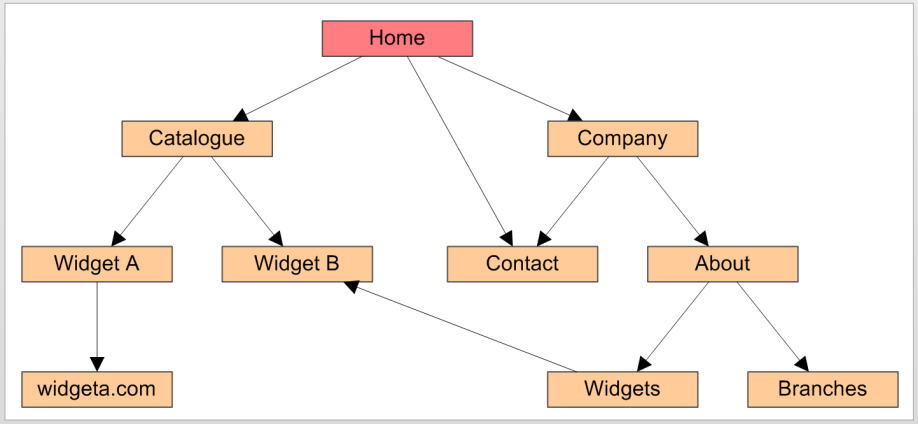
Outline

1. Background
2. Existing Work
3. Approach
4. Results
    1. CASE tool
    2. Model completion
    3. Contributions
5. Future Work

Background

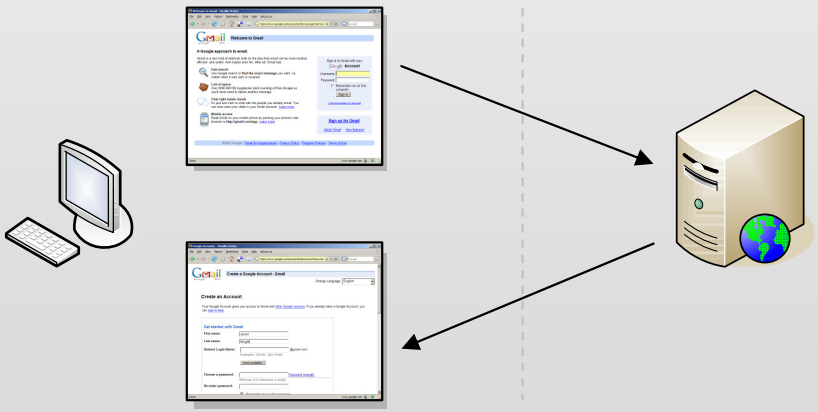
- "Web 0.x": Pages and hyperlinks
- Modelled with simple directed graphs
  - Hyperlinks, <a>

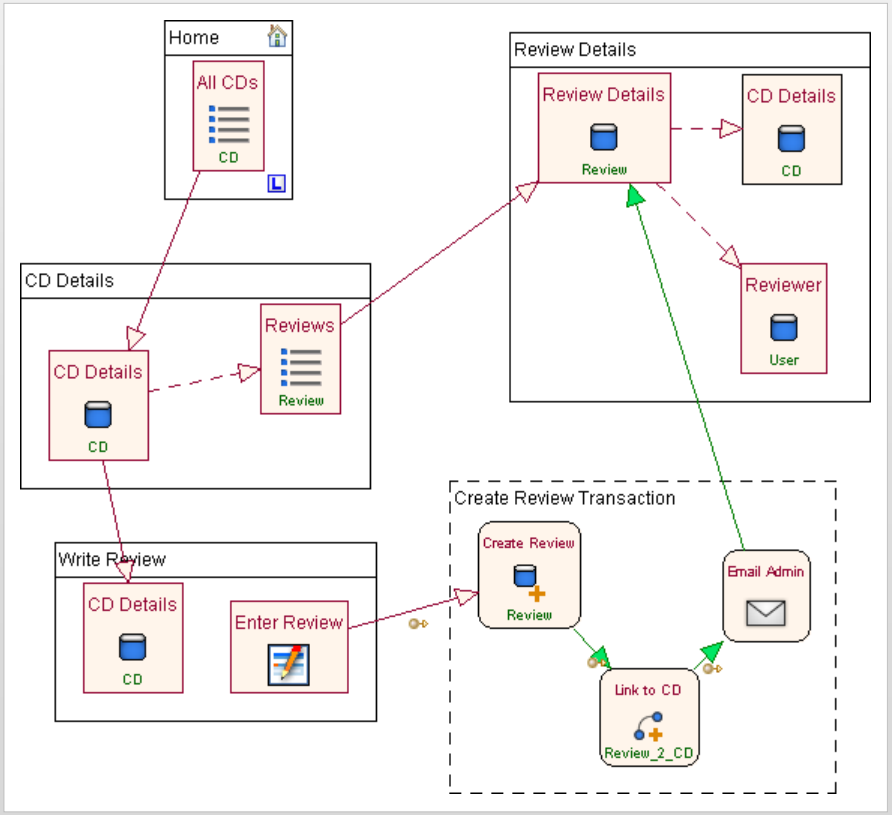




Background

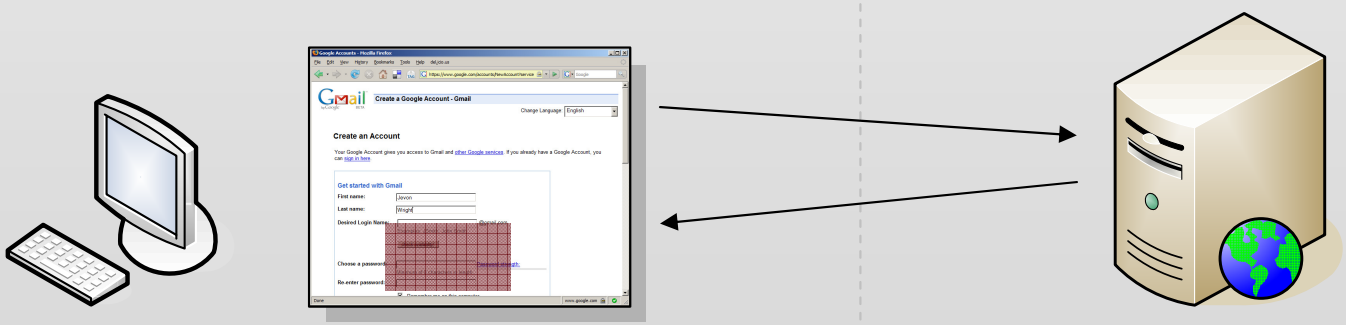
- "Web 1.x": Web applications
- Oriented around databases
  - Request-respond interaction
  - WebML, UWE



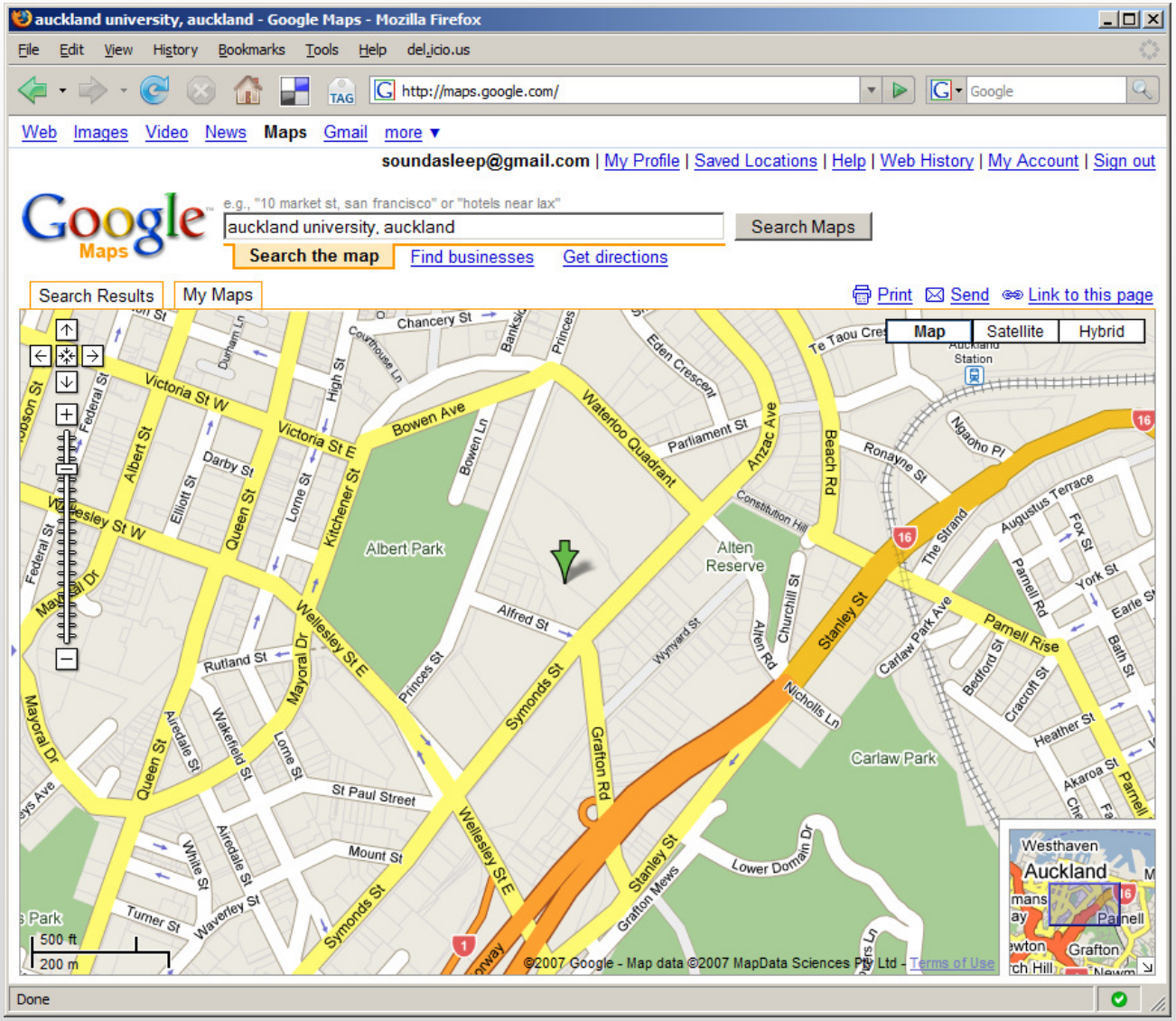


Background

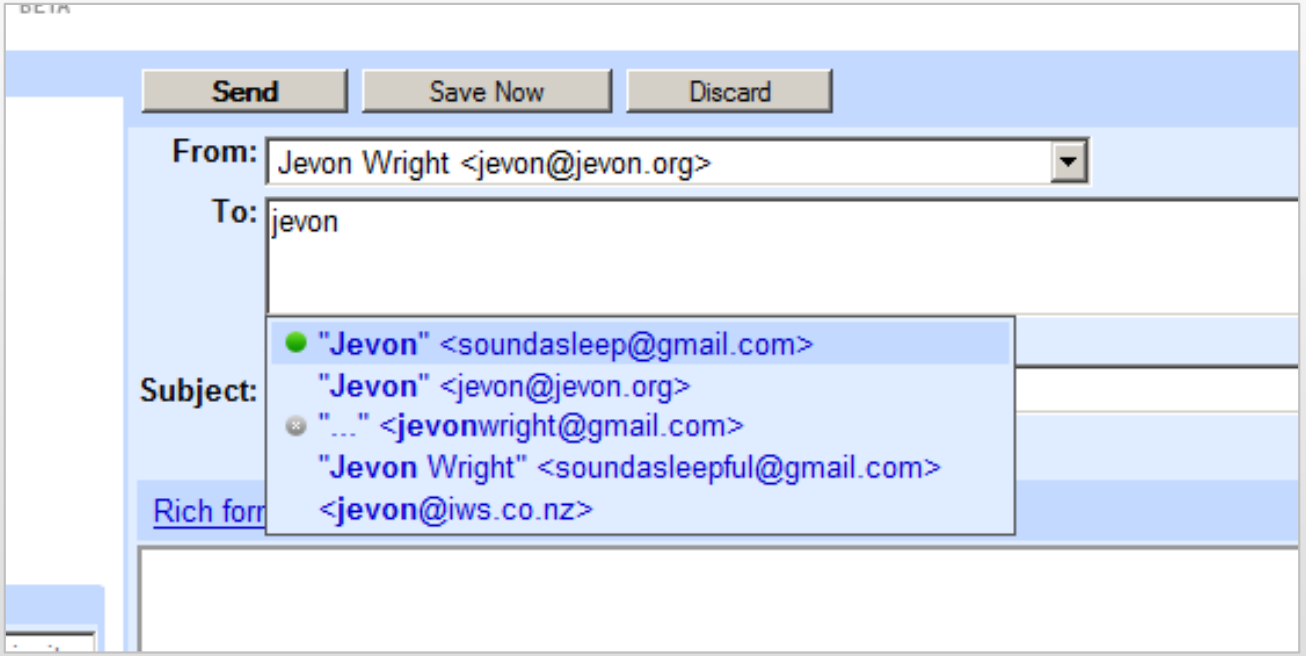
- "Web 2.0": RIAs powered by AJAX
  - Richer user interfaces
  - Client-side Javascript
  - Client-side DOM
  - Background Network Requests



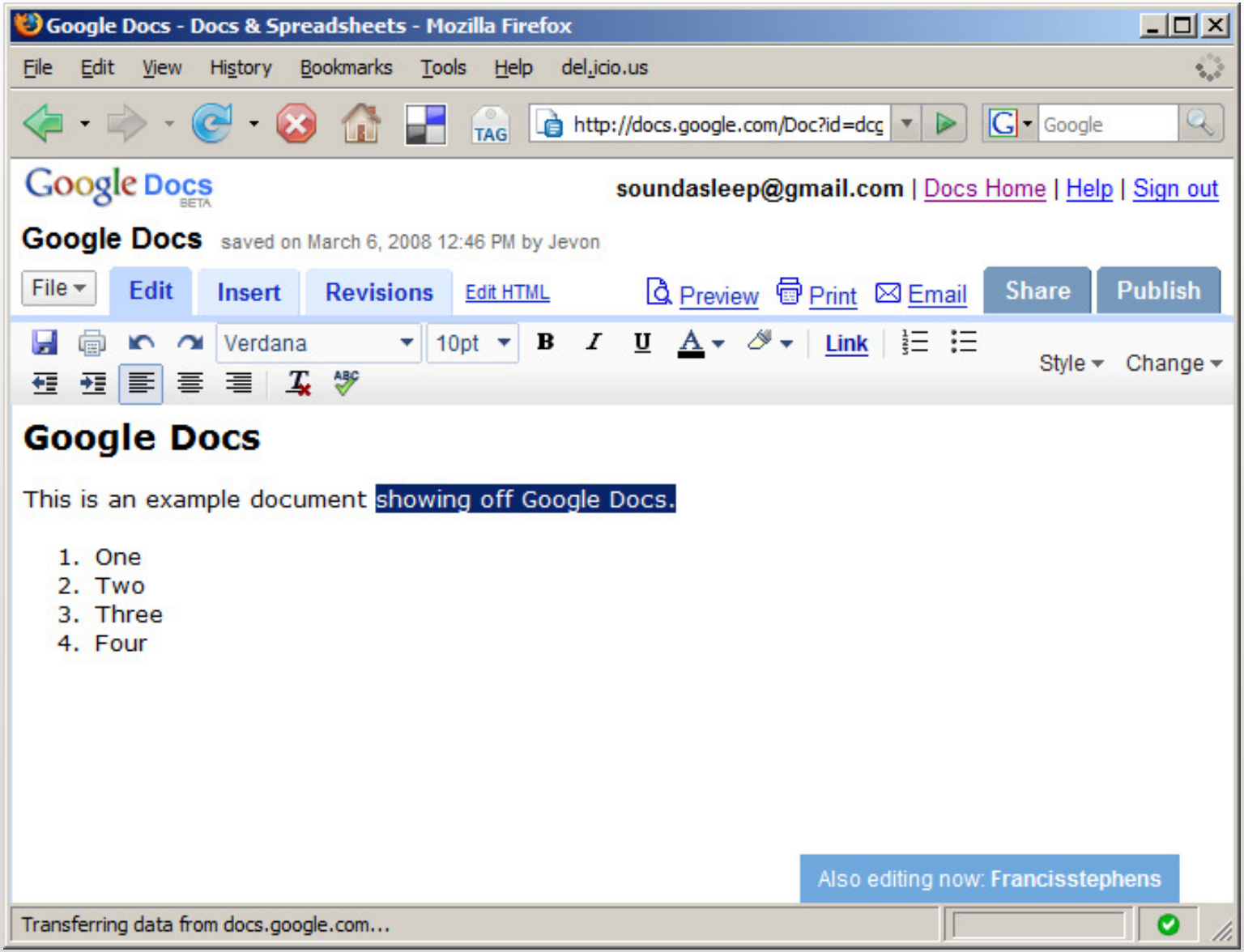
Rich Internet Applications



Rich Internet Applications



Rich Internet Applications



Motivation

- Current Situation
  - Technologies
    - HTML, CSS, PHP, SQL, Javascript, .Net, XML, JSON, ...
  - Platforms
    - Firefox, Internet Explorer, Opera, Safari, ...
  - Devices
    - PC, Console, iPhone, Mobile, ...
  - Requirements
    - Accessibility, security, compatibility, internationalisation, localisation, ...

Thesis

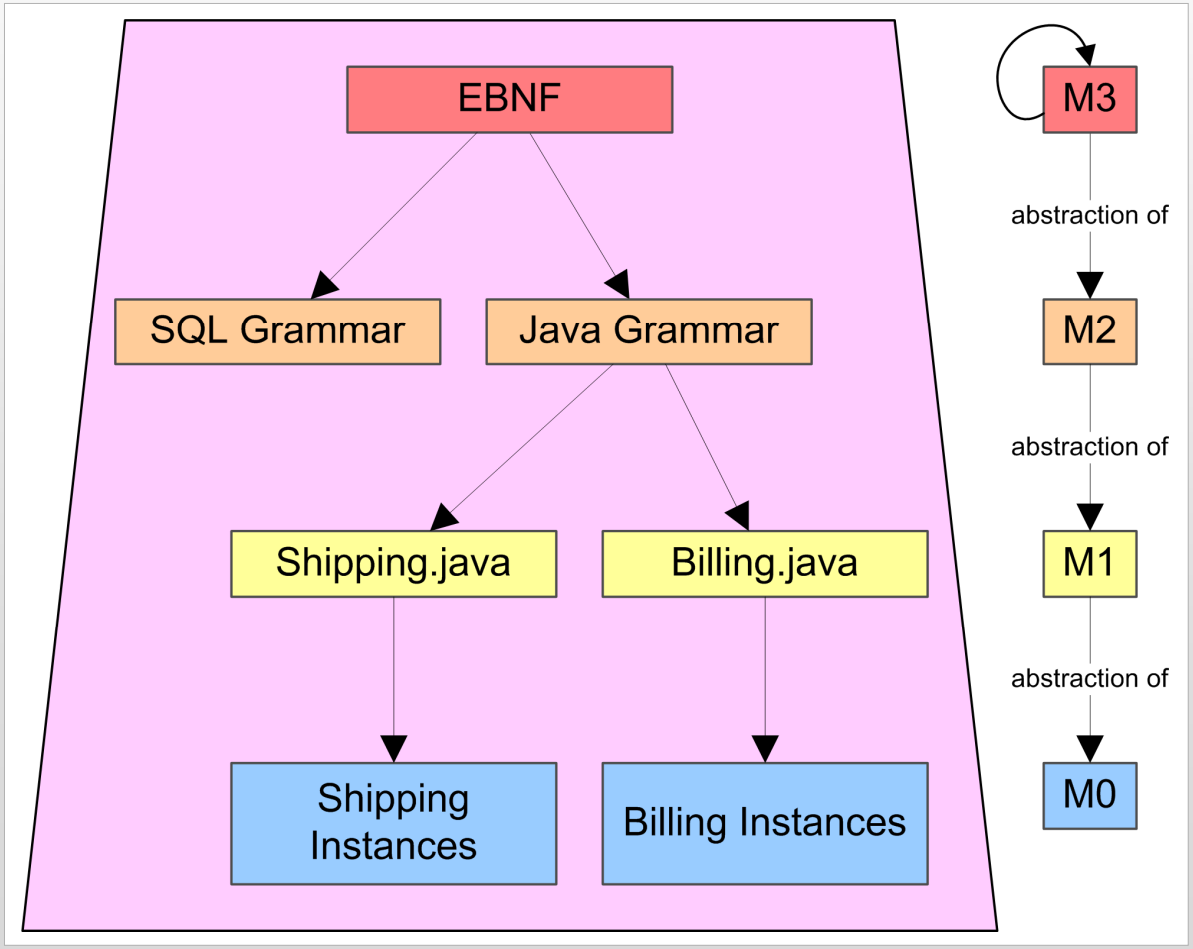
*The development of a new interactive **web application modelling language** will allow us to solve the challenges faced in Web Engineering; in particular, improving the **reliability**, **usability** and **security** of interactive web applications, and **simplifying** the development process in industry.*

Models

- Abstract simplifications of reality
- Common in computing
  - Java software: models of applications
  - HTML: models hypertext
  - Java AWT: models user interfaces
  - UML: models software systems
- Domain-specific languages (DSLs)

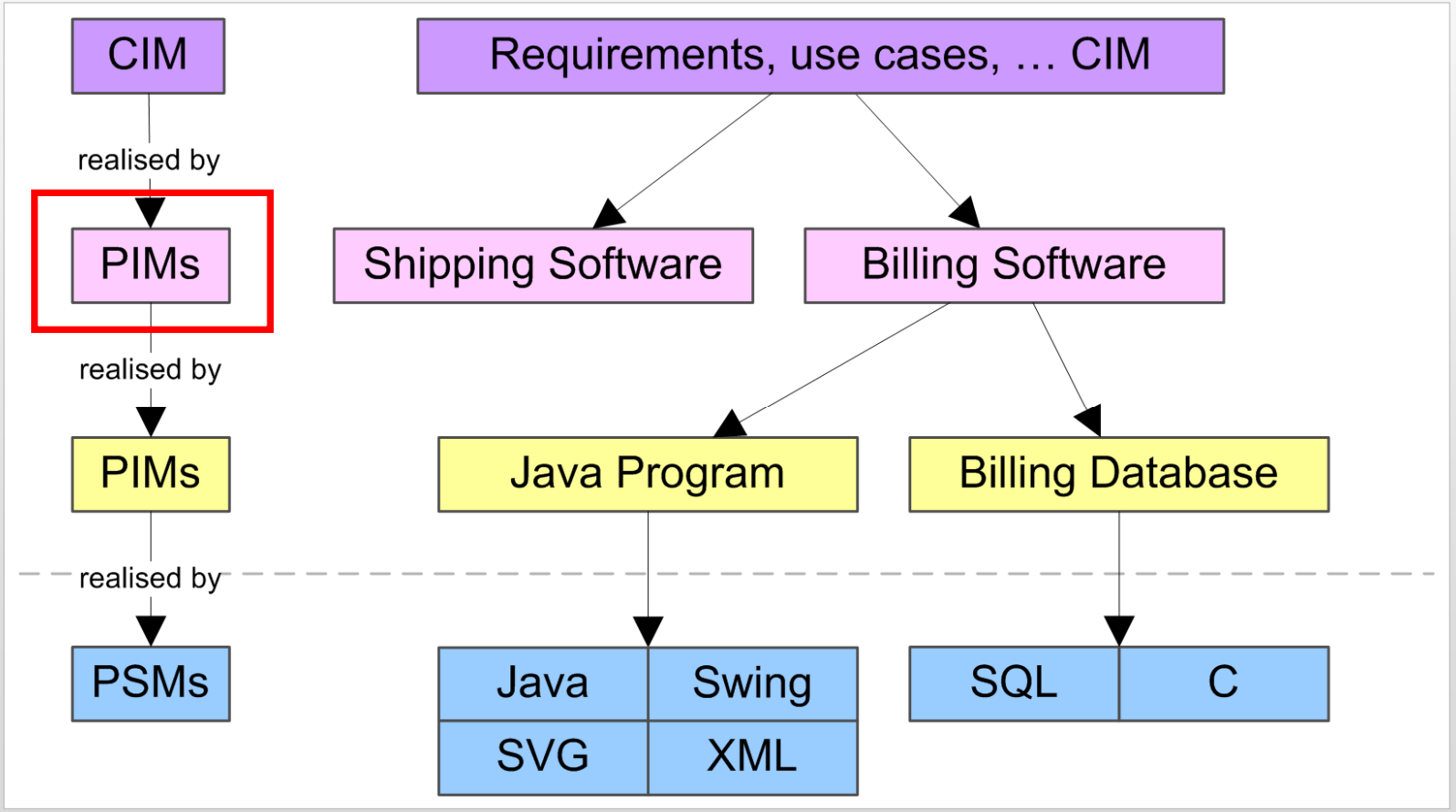
Models

- Meta Object Facility (MOF)



Models

- Model Driven Architecture (MDA)



Models

- Model-driven Development
  - *"Modelling and transforming is a better foundation than programming"*
- Ideal world
  - Develop one platform-independent model
  - Generate to multiple platforms
  - Focus on the design, not the technical side

Nora Koch, "Classification of Model Transformation Techniques Used in UML-based Web Engineering," Software, IET 1(3), 2007.

Requirements of RIAs

- Identifying the requirements of RIAs
- 68 use cases
- 59 core requirements
  - Software support
  - E-mailing users
  - Client-side timers
  - User collaboration
  - Drag and drop
  - Pop-up dialog boxes
  - Controlling the 'back' button

#	Requirement	Example
Interaction		
T1	E-mailing Users	Gmail: Can send e-mails
T2	E-mail Unsubscription	Facebook: User can unsubscribe from all e-mails
T3	Mobile Phone Communication	Calendar: Can send text message reminders
T4	Internationalisation Support	Last.fm: Different locales
T5	Multiple Domain Support	Last.fm: Different domains display different locales
User Interface		
U1	Presentation	Calendar: Displaying a particular user interface
U2	Client-side Scripting	Gmail: Home page displaying available space
U3	Drag and Drop	Calendar: Can drag and drop events
U4	Loading Time Support	Gmail: Switch to HTML view after 30 seconds
U5	Keyboard Shortcuts	Calendar: Can browse using keyboard
U6	Opening New Windows	Pages: Open links in new windows
U7	Pop-up Dialog Boxes	Gmail: Can compose an e-mail in a new window
U8	Runtime Interface Updates	Gmail: Update *Unread Mails* in real time
U9	Static Views (HTML)	Gmail: Provide a static HTML view
U10	Modal Dialogs	Pages: Inserting an image shows a modal dialog
U11	Use External Components	Facebook: Transitions with *script.aculo.us*
U12	Provide External Libraries	Gmail, Calendar: A consistent calendar input box

Table 2. Interactive Web Application Requirements (2)

Wright, J., Dietrich, J., "Requirements for Rich Internet Application Design Methodologies," in Proceedings of WISE 2008, 2008.

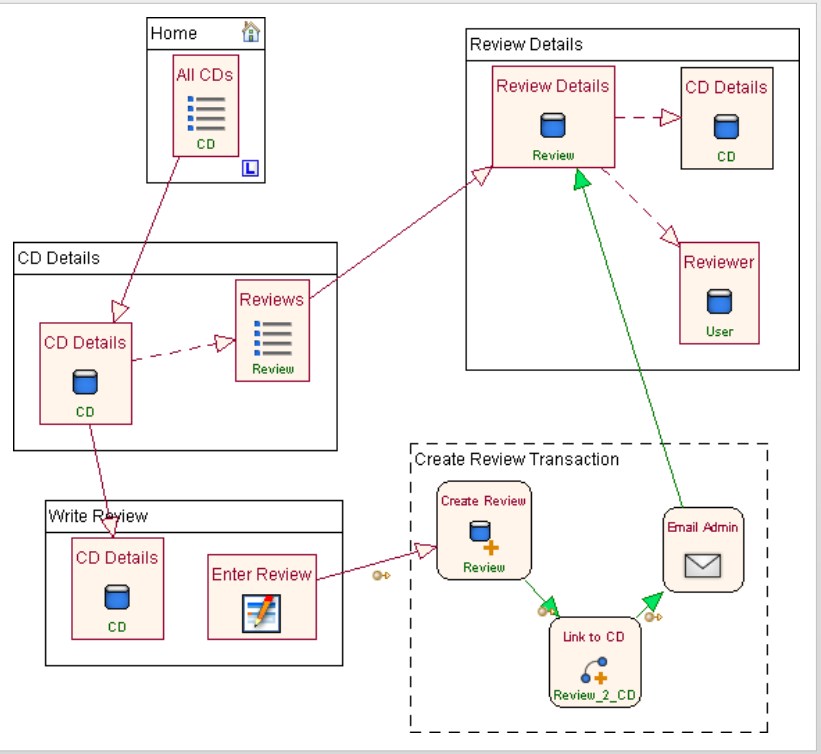
Requirements of RIAs

- *Ticket 2.0*
  - Combines all RIA requirements
  - A useful modelling benchmark
  - Implemented using Symfony (PHP)
  - http://openiaml.org/ticketsf/
- Other domain-specific benchmarks
  - EURent for rules
  - Java Pet Store for EE
  - Smart Ticket for Mobile Apps

Wright, J., Dietrich, J., "Requirements for Rich Internet Application Design Methodologies," in Proceedings of WISE 2008, 2008.

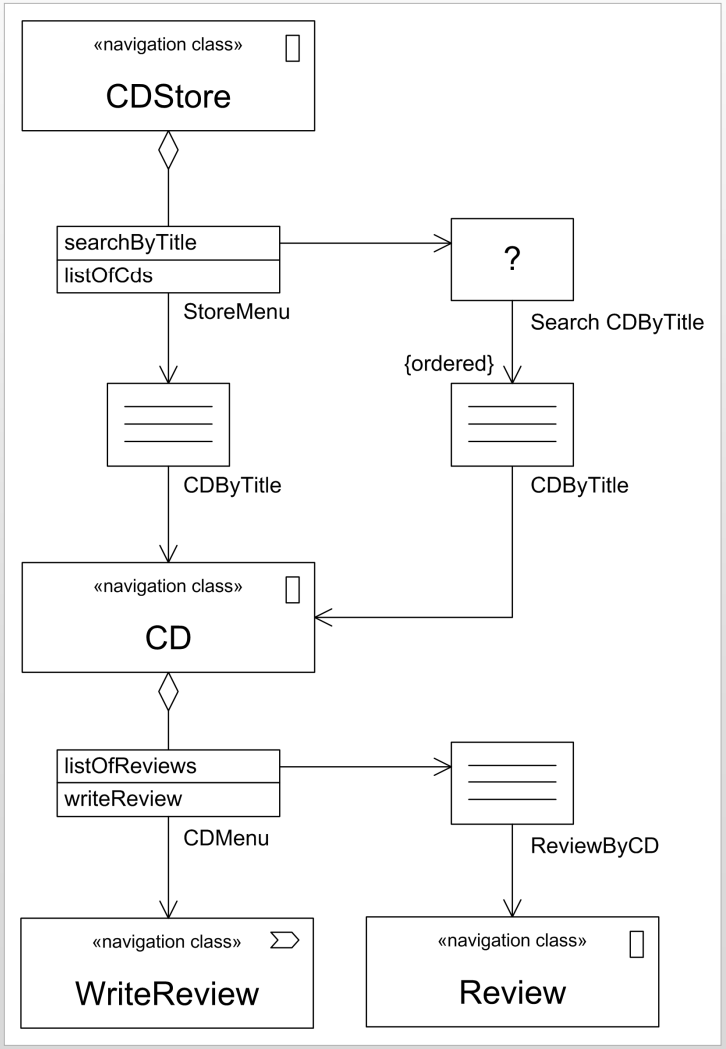
Existing Work

- WebML
  - Proprietary (initially DTDs, now EMF)
  - Five models
  - Case tool: WebRatio
  - Mostly only server-side
  - Suite of client-side components (2006)
  - RUX-method adds client-only interactivity (2007)



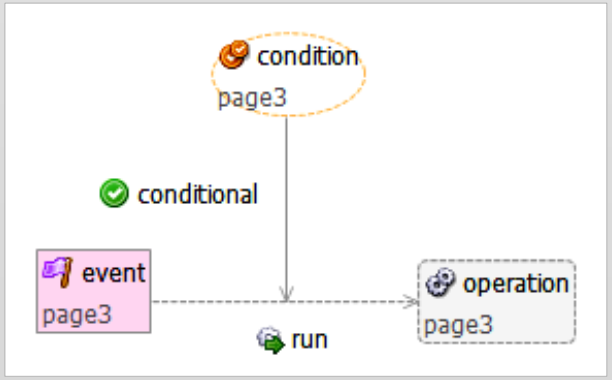
Existing Work

- UWE
  - Extension of UML
  - 9+ models, following MDA
  - Model transformations
  - Server-side only
  - RUX-method adds client-only interactivity (2008)



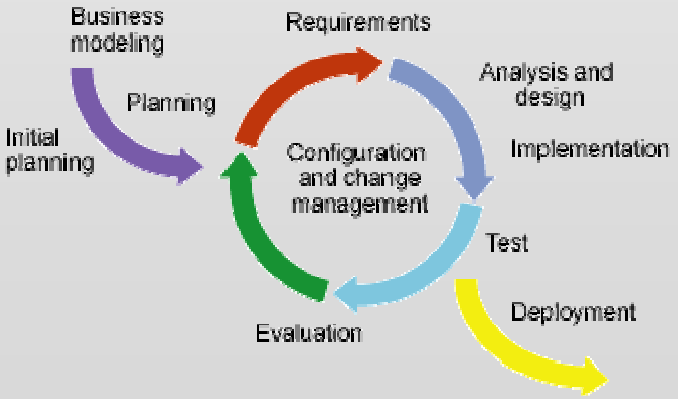
The IAML Model

- *Internet Application Modelling Language*
  - Designed to support RIA concepts **directly**
  - Based on existing models:
    - UML activity diagrams, class diagrams
    - Role-based access control
- First-class Concepts
  - Event/condition/action rules (ECA)
  - Domain objects, users
  - Security, authentication



Approach

- Constantly changing requirements
  - New business models
  - New standards (HTML5, Local Storage, SVG, Canvas)
  - New browsers (Firefox 3.5, IE 8)
  - Google Gears, Prism, Silverlight, AIR
- Iterative development
  - Focus on deliverables
  - Test-driven development



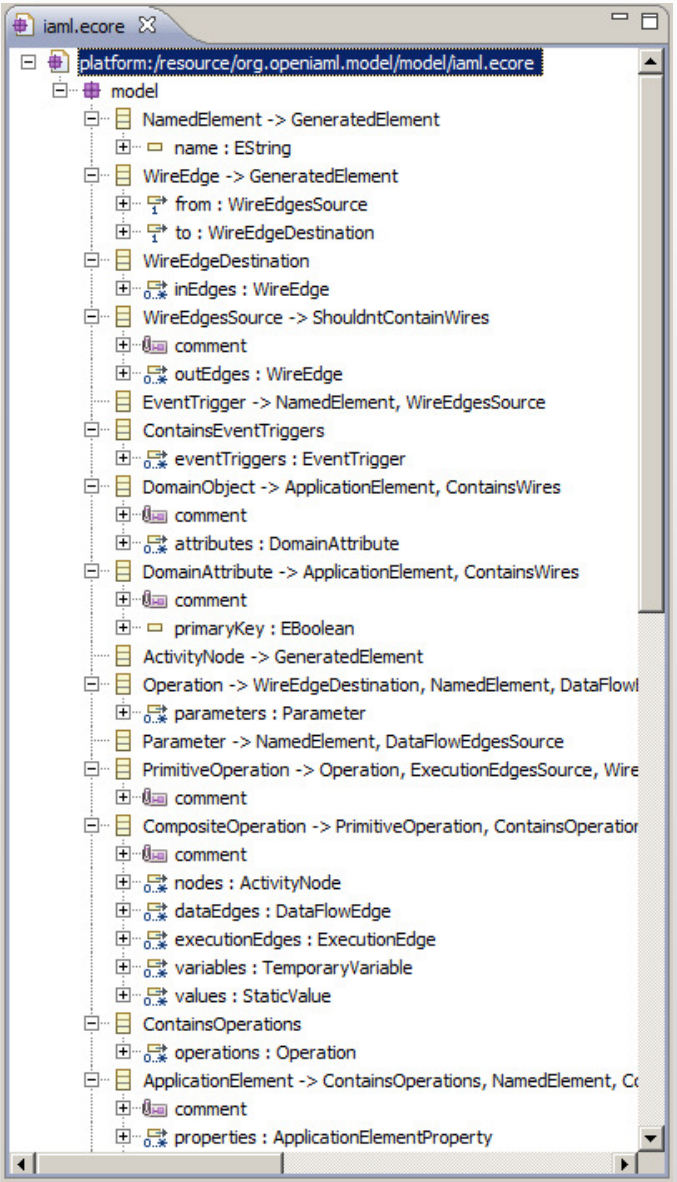
Milena Litoiu, “Reduce complexity with model-driven development, Part 1,” IBM, Tech. Rep., 2004. [Online]. Available: http://www.ibm.com/developerworks/ibm/library/i-modev1/

CASE Tool

- CASE Tool
  - Proof-of-concept
  - Ongoing work
  - Version 0.4 (October 2009)
- Implemented in Java
  - Eclipse framework
  - EMF meta-modelling framework

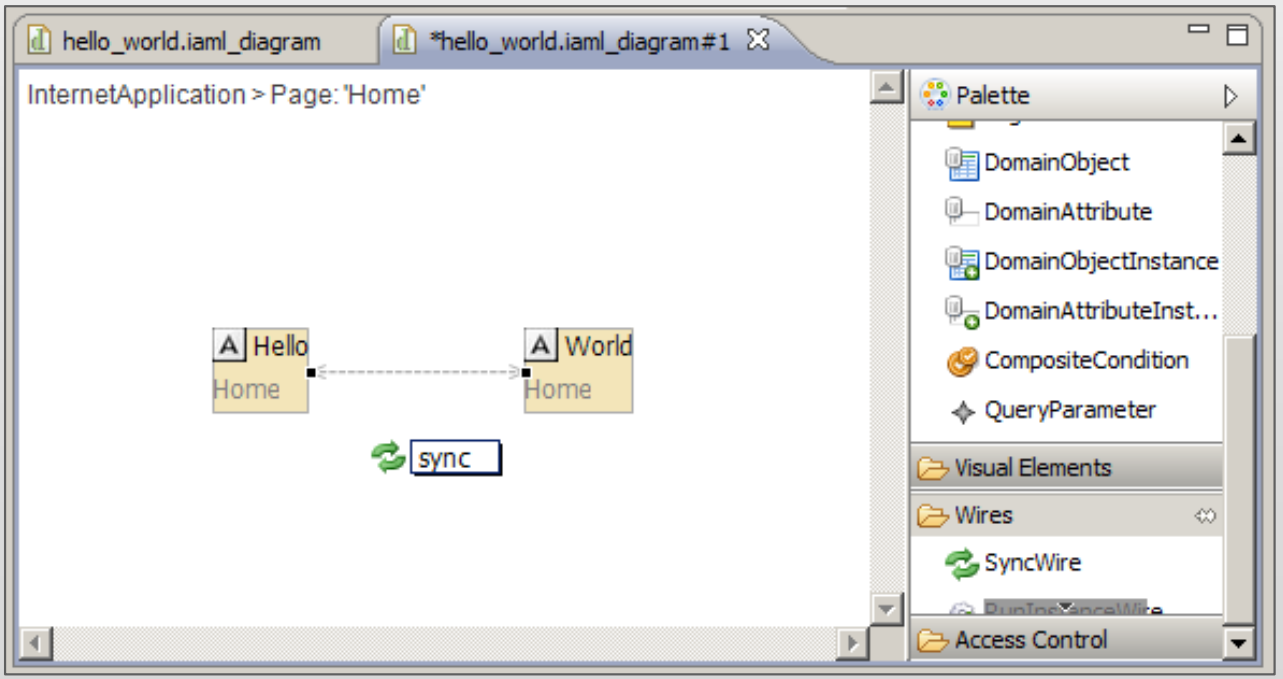
CASE Tool: Metamodel

- Eclipse Modeling Framework (EMF)
  - MOF-compatible
  - Define the meta-model
  - Generates Java model code
  - Easy integration
- Version 0.4
  - 83 classes, 8 packages



CASE Tool: Graphical Editor

- Graphical Modeling Framework (GMF)
  - Graphical editing tools
  - Based on EMF
  - Supported by industry (Eclipse, IBM)



CASE Tool: Code Generation

- openArchitectureWare, XPand
  - Based on EMF
  - QVTo implementation
  - Used to generate GMF editors

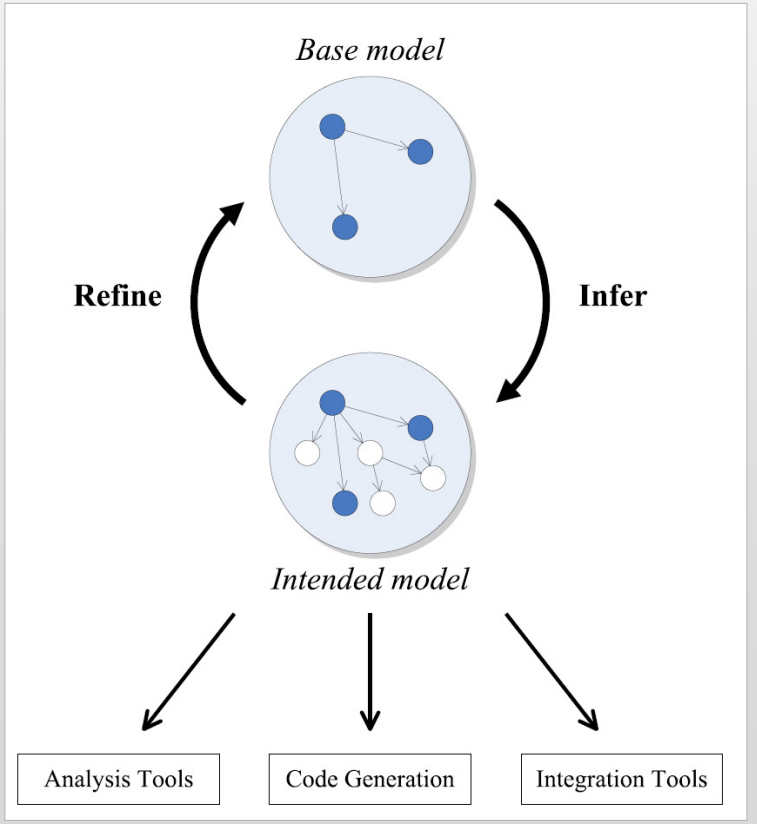
```
«DEFINE expandPage FOR model::visual::Page-»
     «FILE "output/"+safeNameString(id)+".php"-»
          <?php /* generated page "«name-»" */ ... ?>
          <html>
          <head><title>«pageTitle()»</title></head>
          <body>
               <h1>«pageTitle()»</h1>
               «EXPAND expandChild FOREACH children-»
          </body>
          </html>
     «ENDFILE»
«ENDDEFINE»
```

Model Completion

- Good models are difficult to find
  - Detailed: flexible, but big models
  - Abstract: simple, but hard to customise
  - A balance is required
- Model Completion
  - Based on frameworks
  - Design an initial model
  - Infer the completed model

Model Completion

- Model Completion
  - Create an "intended model" from sensible defaults
  - Non-monotonic inference
  - Iterative refinement
  - Maintenence
  - Modify generated models
- Submitted to ASWEC (2010)



Model Completion

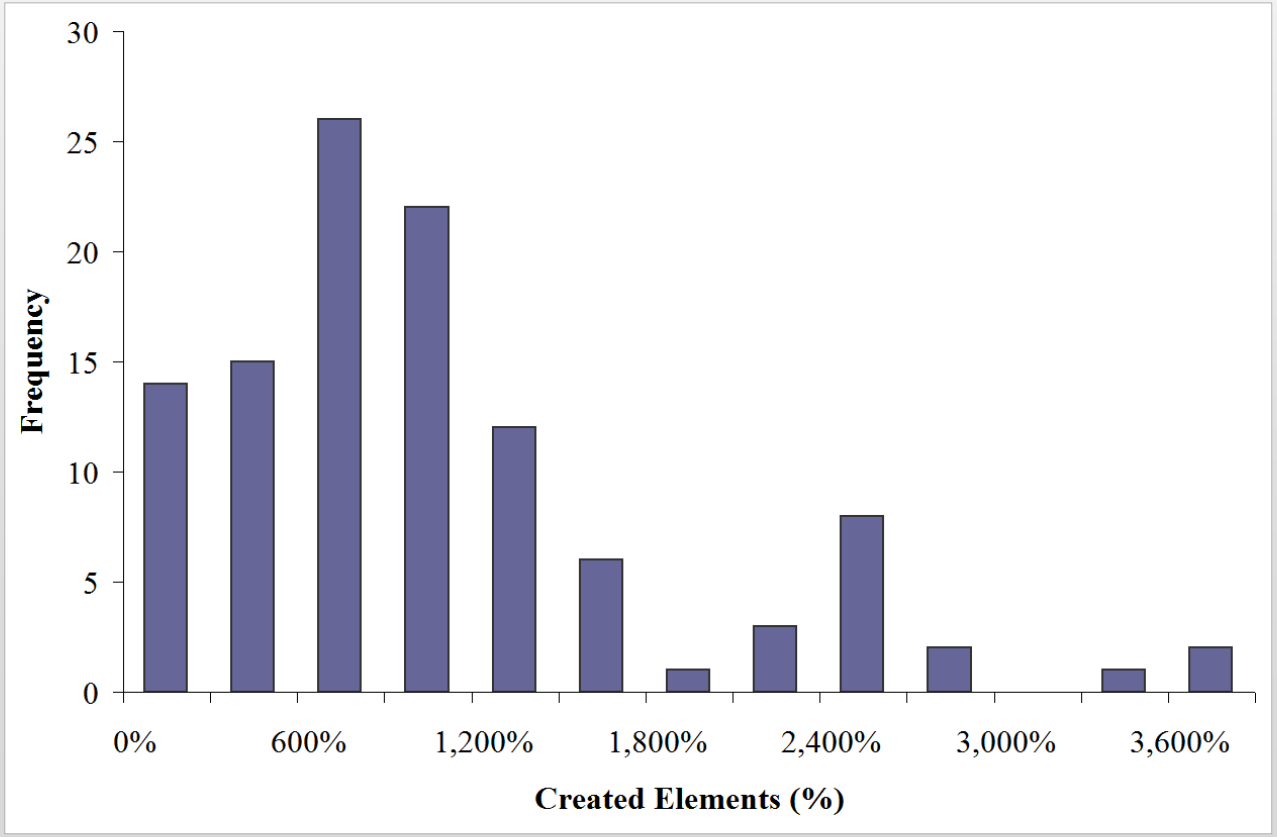
- Implemented in JBoss Drools
  - Suite of 131 model completion rules
- Example
  - Text Fields should have an 'update' operation

```
rule "Create 'update' operation for input text field"
when
     f : InputTextField( overridden == false )
     not (Operation( eContainer == f, name == "update" ))

then
     CompositeOperation operation =
         handler.generatedCompositeOperation(f, f);
     handler.setName(operation, "update");
     cache.add(operation, drools);
end
```

Model Completion

- Evaluated against 112 test models
  - Average of 1,100% new elements created



Progress

- IAML 0.4 (October 2009)
- Modelling constructs
  - Operations (UML activity)
  - Databases (UML class)
  - Events and conditions (ECA rules)
  - Database querying and instancing
  - Role-based access control
  - User authentication
  - Limited navigation and UI modelling

Example

- Example
  - Limiting access to Registered Users

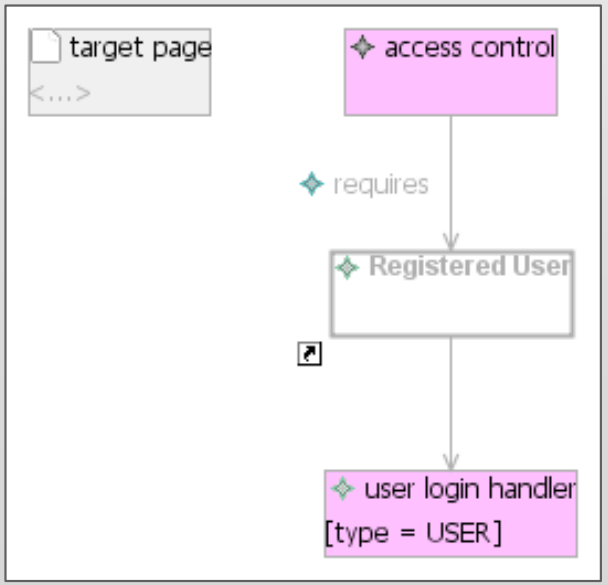
- http://iaml.googlecode.com/svn-history/r1310/trunk/examples/UserRolesLoginHandler/UserRolesLoginHandler.html

Example

- Define the Application

- Define the Users

- Restrict Access



Design Verification

- Design Verification
  - Current Work
  - Security: prevent unauthorised data access
  - Accessibility: across platforms and devices
  - Concurrency
  - Difficult if analysing source code
    - *"Unfeasible to extract structural design intent from code"*
  - Examples: Infinite loops, XSS vulnerabilities

Massoni, T., Gheyi, R., Borba, P., "A Model-Driven Approach to Formal Refactoring," in OOPSLA '05, 2005.

Design Verification

- Approach Requirements
  - Expressive
  - Formal semantics (security)
  - Traceable
  - Scalable, fast
  - Easy integration
- Approach?
  - Static analysis (OCL, CrocoPat)
  - Model checking (Alloy, NuSMV)

Future Work

- Complete the meta-model (focus)
  - Rich data types
  - Sending e-mails
  - UI modelling
    - Drag/drop, keyboard shortcuts, media, ...
  - Scheduled events (client, server)
  - Input validation
  - Concurrency (transactions, rollback, ...)
  - ...

Future Work

- Evaluation against *Ticket 2.0*
  - Model complexity metrics
- Platform independence
  - Implementation in JSP
  - Mobile devices
  - Dealing with Internet Explorer

Future Work

- Extensibility
  - Eclipse is built upon Plugins
  - Potentially extensible:
    - Eclipse editors
    - EMF meta-model
    - GMF editors
    - Drools rules
    - Code generation templates
    - Design validation rules
  - Packaging components into features?

Success

- Important
  - Can model *Ticket 2.0*
  - Can generate web applications
  - Supporting CASE tool
- Desirable
  - Can generate *Ticket 2.0*
  - Multiple platforms
  - Model verification
  - Community feedback

Contributions

- Published work
  - Survey of existing languages (APCCM 2008)
  - Identifying RIA requirements (WISE 2008)
- Submitted work
  - Model completion (2010?)
- Future work
  - Design verification?
  - Evaluation against *Ticket 2.0*

Conclusion

- IAML 0.4 available online
  - Eclipse plugin
  - Free and open source (EPL)

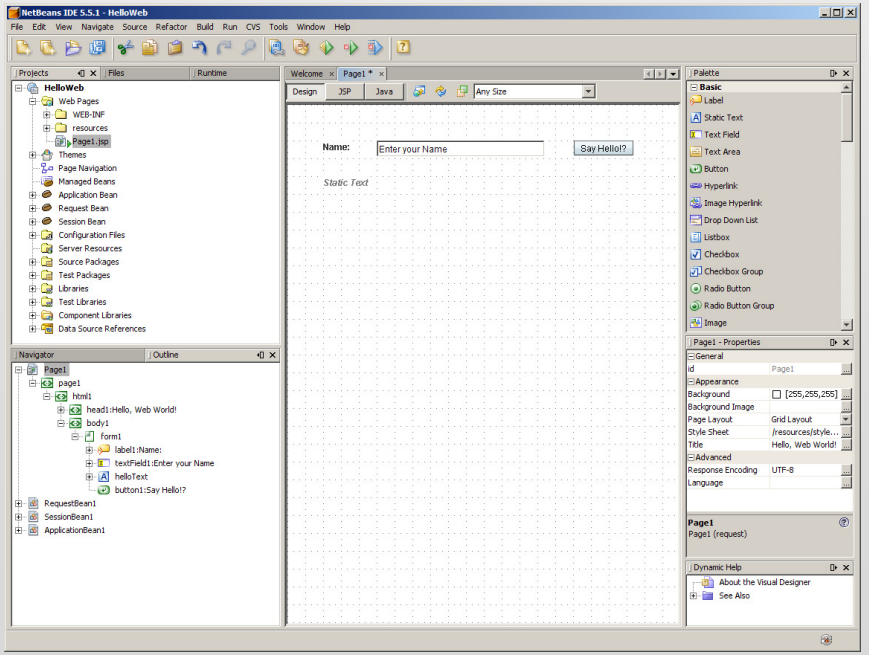
j.m.wright@massey.ac.nz

http://openiaml.org

Additional Slides

Existing Work

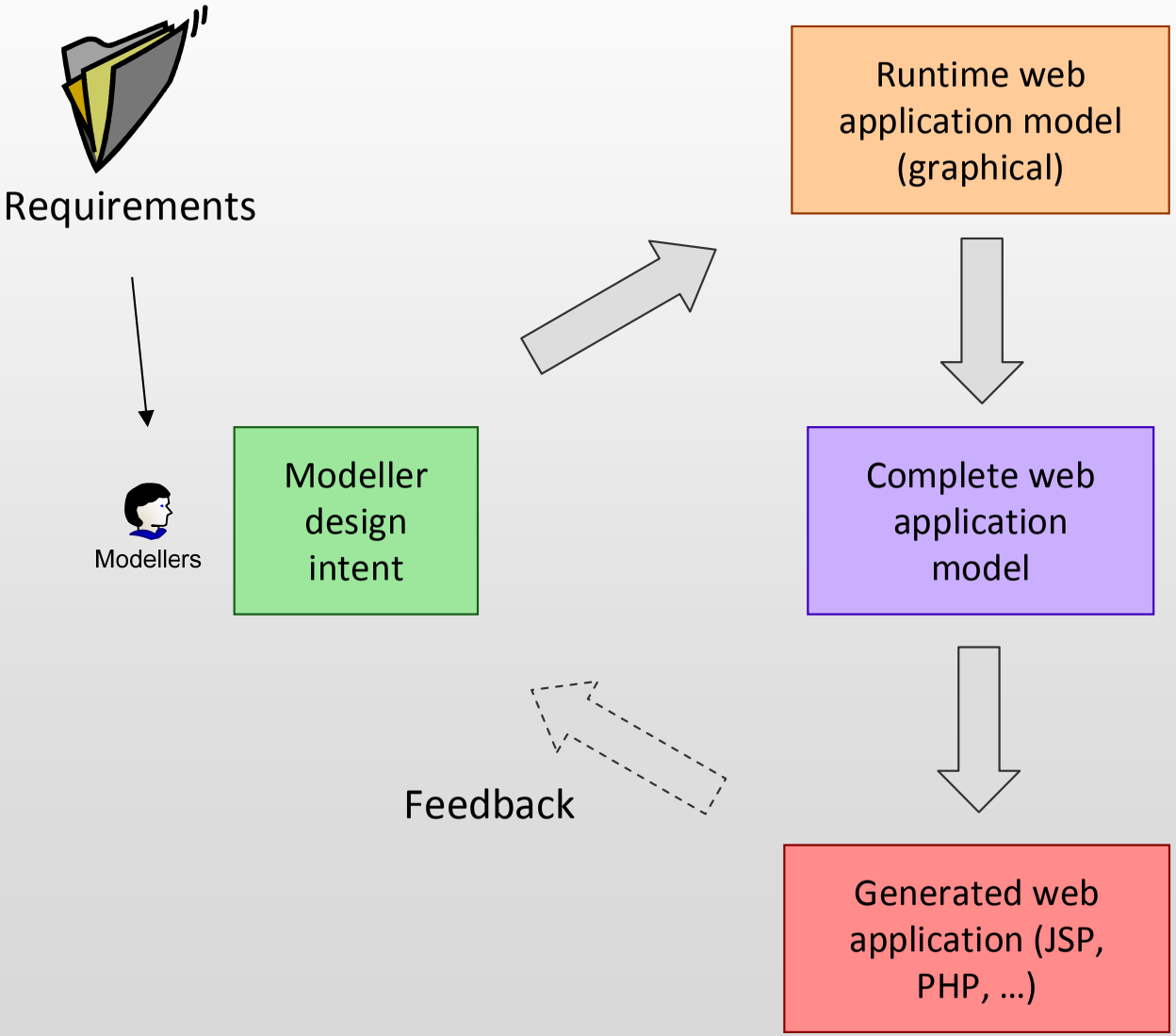
- Commercial Approaches
  - NetBeans
  - ASP.Net
  - Dreamweaver
  - Skyway Perspectives
- Frameworks
  - Symfony
  - Ruby on Rails
  - Spring



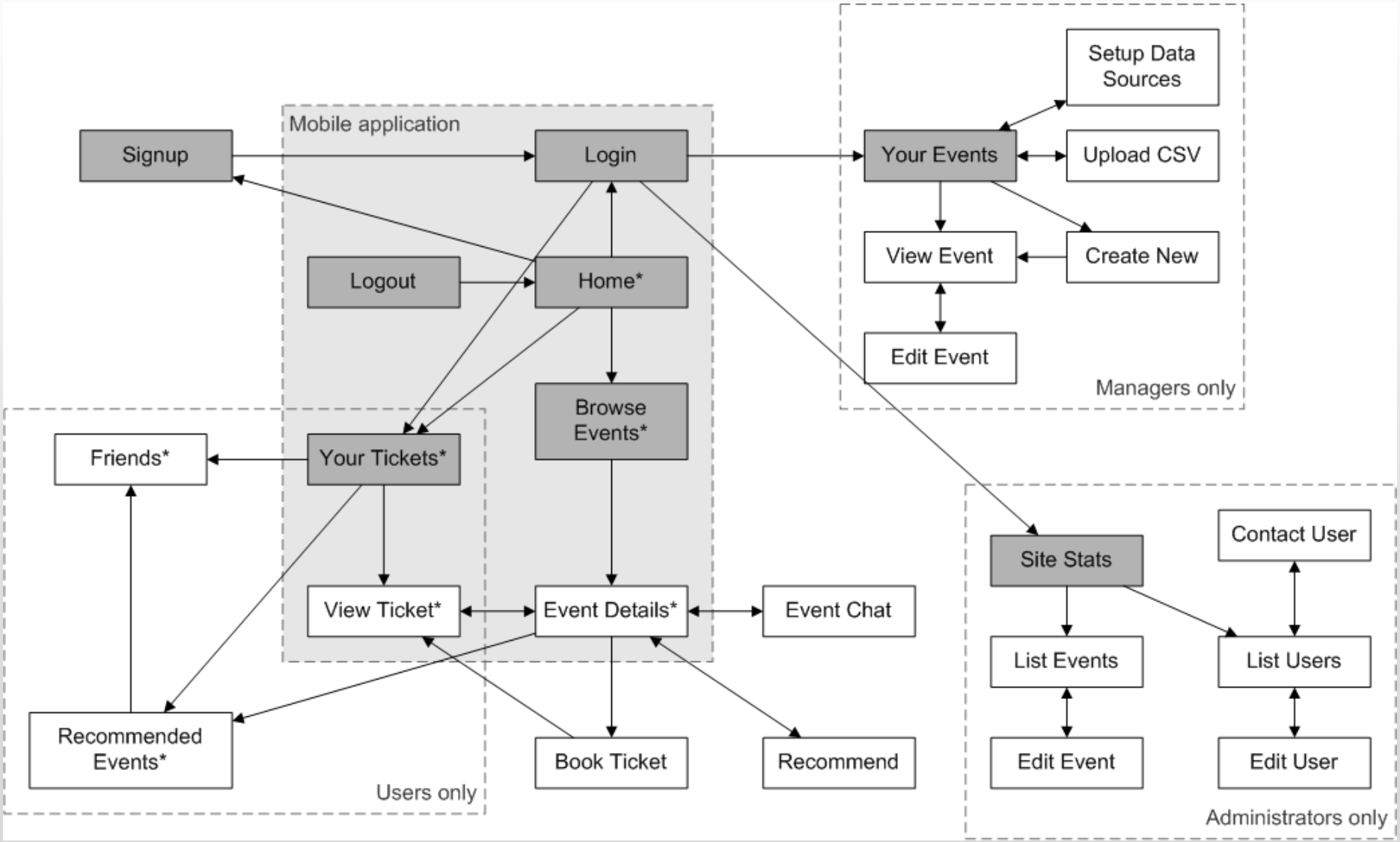
Existing Work

- Other UML approaches
  - WUML
  - Conallen
  - UML doesn't (yet) have declared semantics
- UML could definitely model RIAs
  - But we need an executable approach
  - Our approach can be integrated into UML

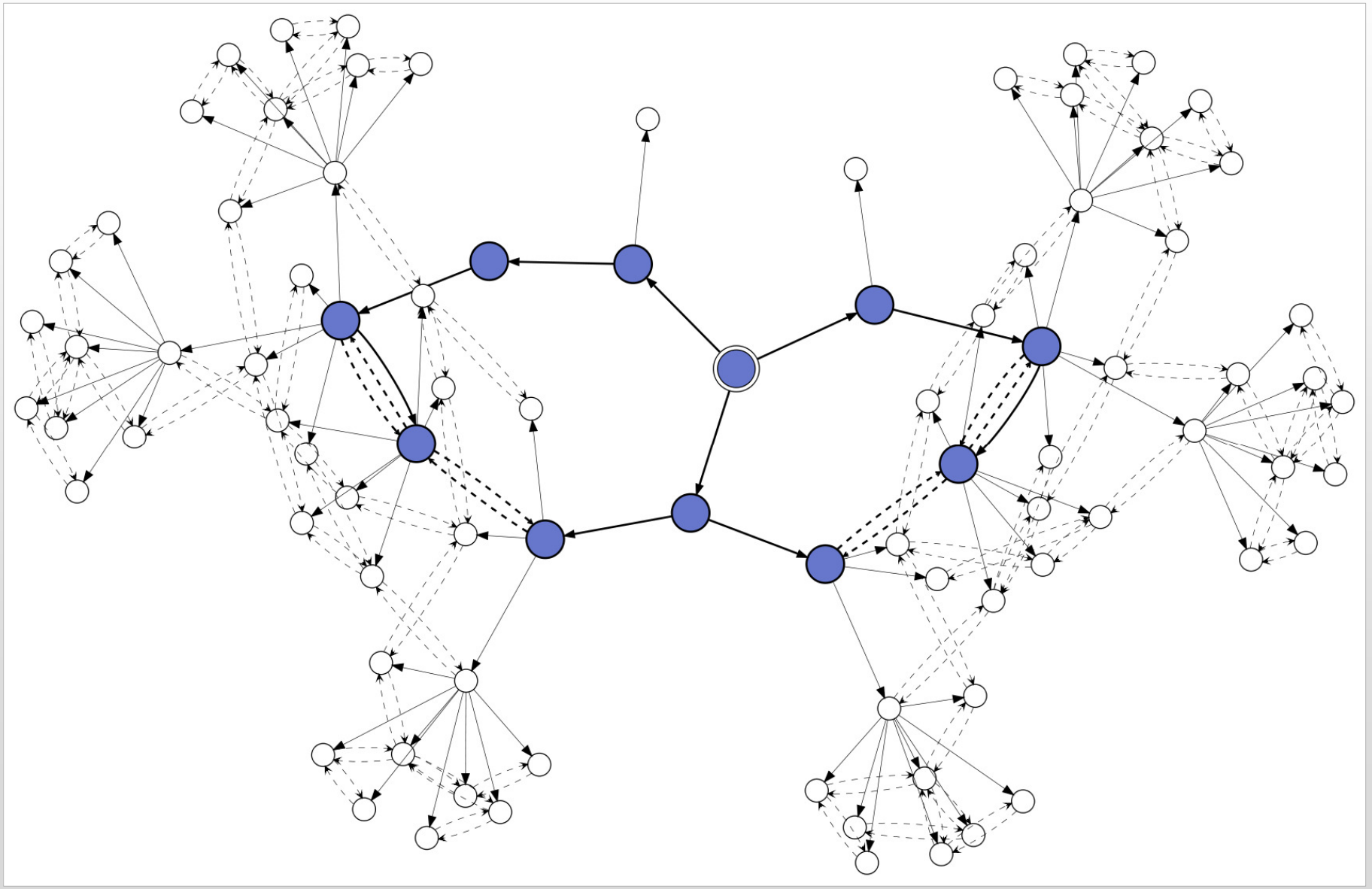
CASE Tool Overview



Ticket 2.0


Mobile application
Signup
Login
Logout
Home*
Browse Events*
Your Tickets*
Friends*
View Ticket*
Event Details*
Event Chat
Recommended Events*
Book Ticket
Recommend
Users only
Setup Data Sources
Your Events
Upload CSV
View Event
Create New
Edit Event
Managers only
Site Stats
Contact User
List Events
List Users
Edit Event
Edit User
Administrators only

Model Completion



Model Completion