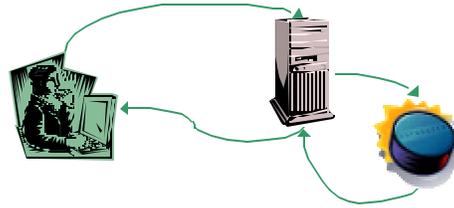
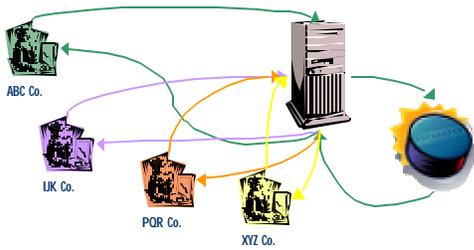


Implementing Model-View-Controller (MVC) with Fusebox

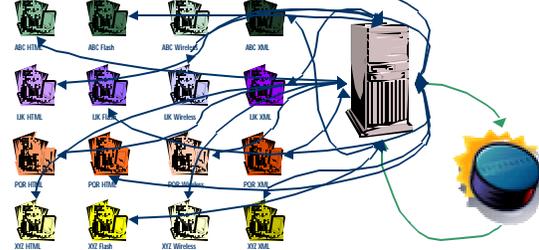
"Let's keep things simple!"



Keeping it simple?



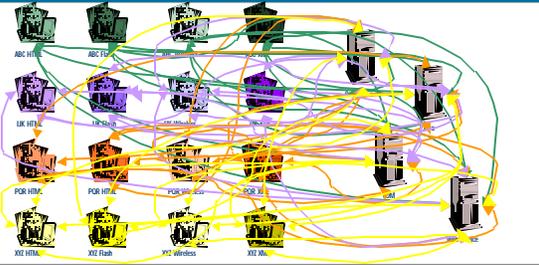
This isn't so simple...



But wait...what about the back end?



It's pasta for everybody!!



In pace requiescat...



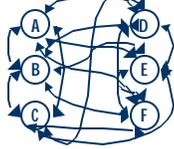
A chat with the Great Webmaster

3 components : 3 unique interactions



The problem of complexity

6 components : 15 unique interactions



50 components : 1225 unique interactions

A picture of a cat



MVC: What is it?

- MVC is a software architecture originating in the Smalltalk world.
- The modeling of the external world and the visual feedback to the user are explicitly separated and handled by three different components, each specialized for its task.
- Sun endorses a modified version of MVC for their "Model 2" architecture.

MVC components: Model

- The *model* is responsible for representing the external world in an abstracted or idealized state. Algorithms and databases belong to the model.

MVC components: View

- The *view* is responsible for displaying information from the user and retrieving information from the user. In the web world, this is normally done using HTML, with Flash and Java used when greater flexibility is desired.

MVC components: Controller

- The *controller* interprets mouse and keyboard inputs from the user, commanding the model and/or the view to change as appropriate.

This is the model



This is the view



This is the controller



Model or View?

- Fuseactions related to a USER
 - `addUser` displays a form for entering new user info

Implementing MVC with Fusebox

Model or View?

- Fuseactions related to a USER
 - **addUser** displays a form for entering new user info: VIEW

halhelms

Implementing MVC with Fusebox

Model or View?

- Fuseactions related to a USER
 - **editUser** displays a form with user info filled in

halhelms

Implementing MVC with Fusebox

Model or View?

- Fuseactions related to a USER
 - **editUser** displays a form with user info filled in : MODEL to get information from database; VIEW to display info

halhelms

Implementing MVC with Fusebox

Model or View?

- Fuseactions related to a USER
 - **deleteUser** removes a record from the database

halhelms

Implementing MVC with Fusebox

Model or View?

- Fuseactions related to a USER
 - **deleteUser** removes a record from the database : MODEL

halhelms

Implementing MVC with Fusebox

Model or View?

- Fuseactions related to a USER
 - **listUsers** returns a list of all users

halhelms

Model or View?

- Fuseactions related to a USER
 - **listUsers** returns a list of all users : MODEL

Model or View?

- Fuseactions related to a USER
 - **loginUser** displays a login form

Model or View?

- Fuseactions related to a USER
 - **loginUser** displays a login form : VIEW

Model or View?

- Fuseactions related to a USER
 - **validateUser** checks the info submitted against a DB

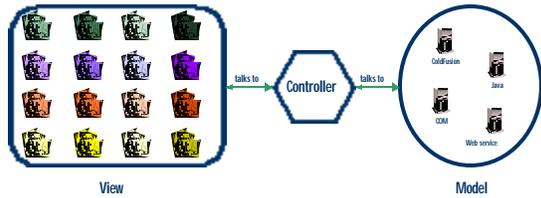
Model or View?

- Fuseactions related to a USER
 - **validateUser** checks the info submitted against a DB : MODEL

Model or View?

- Fuseactions related to a USER
 - **insertNewUser** inserts information into a database

Restoring sanity with MVC



Meet Vinny

- Vinny is the proprietor of a social club in New York City, specializing in games members can play.
- As part of his organization, Vinny needs our help in keeping track of his employees.
- Let's help Vinny out...

A Fusebox primer

- *Discovering Fusebox 3*: www.halhelms.com
- All actions in a Fusebox application begin as requests made of the application, known as *fuseactions*.
- *Circuits* are groupings of related fuseactions and correspond to file directories.
- Circuits can be nested to reflect parent-child relationships.

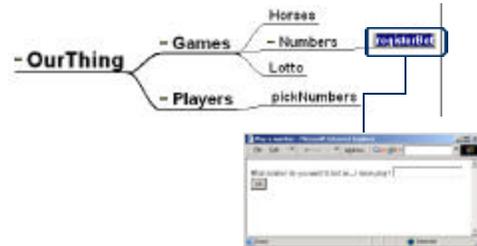
A Fusebox primer

- A fully qualified fuseaction includes the circuit name and the fuseaction request in the following format:
 - circuit.fuseaction_request
- Examples:
 - Login.validate
 - Cart.addItem
 - User.new

A Fusebox primer

- *Fuses* are individual code files that work either singly or (more commonly) in conjunction with others to fulfill a fuseaction request.
- Example fuses:
 - act_ValidateLogin.cfm
 - qry_AddItemToCart.cfm
 - dsp_UserForm.cfm

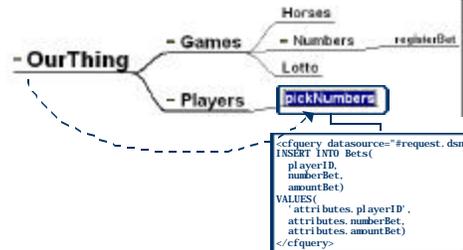
Fusebox in practice: Numbers.registerBet



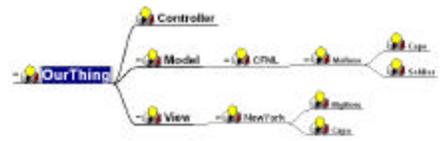
Fusebox in practice: Players. pickNumbers



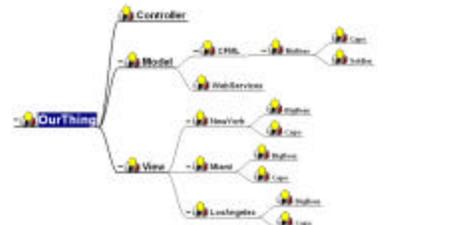
Fusebox in practice: completing Players. pickNumbers



Back to Vinny: implementing MVC



Benefitting from MVC



For more about MVC...

- Online resources:
 - www.halhelms.com
 - www.techspedition.com
 - www.cs.indiana.edu/~cbarav/projects/mvc.html
 - java.sun.com/blueprints/patterns/i2ee_patterns/model_view_controller/
 - www.martinfowler.com/ja/mvc.html
- Training: www.halhelms.com
 - Hal Helms' FastTrack to Fusebox class
 - Hal Helms' Fusebox Mastery class

For more about Fusebox...

- Online resources:
 - www.fusebox.org
 - www.halhelms.com
 - www.techspedition.com
 - www.secretagents.com
- Books
 - *Discovering Fusebox 3* by Hal Helms and John Quarto-vonTivadar (www.techspedition.com)
 - *Fusebox: Developing ColdFusion Applications* by Nat Papovich and Jeff Peters (www.amazon.com)