

Objective	I am seeking consulting or contract projects in software design and implementation, web application development and deployment or system/architecture design in an exciting and challenging work environment that supports full-time telecommuting and encourages creativity, personal initiative, and hard work.
Education	<p>Tufts University, Cambridge, MA (1999-2000) Pursued Masters Degree in EE/CS. Cumulative Average: A.</p> <p>Amherst College, Amherst, MA (1991-1995) Graduated Magna Cum Laude with a B.A. in Computer Science, May 1995. Wrote thesis on Multistrategy Learning Systems. GPA: 3.25/3.81 (Overall/In Major)</p> <p>Andover Academy, Andover, MA (Summer 1989) Studied Basic Electronics and English during six-week summer enrichment program. Cumulative Average: A-</p> <p>Steamboat Springs High School, Steamboat Springs, CO (1986-1990) Graduated Salutatorian with 4.19 GPA (weighted 4-point scale)</p>
Skills	<p>Areas Of Interest And Expertise:</p> <p>Web Frameworks, Web Application Programming, Web/Application Architecture Design & Development, Web 2.0 Design, Strategic/Business Design & Planning, Web Design & Layout, eCommerce Systems Design & Development, Ruby on Rails Technologies, Java Programming, PHP Programming, Java Servlet/Portlet Development, Database Programming/Design, Graphic Design, Client/Server/n-tier Development, Security, CORBA, Distributed Computing, Collaborative Computing, Graphical Applications and User Interfaces, Scripting, CGI Programming, Multimedia Productions, Image Processing, AI Programming</p> <p>In-Depth Knowledge Of:</p> <p><i>Languages:</i> Ruby, PHP, Perl, HTML, DHTML/CSS, JavaScript, JS Libraries, Java, JSP, JFC/Swing, Crypto/Security/X.509, RMI, Jini, Servlets, JDBC, Beans, EJB, C++, C, SQL, CORBA, LDAP, and Pascal</p> <p><i>Architectures:</i> Ruby on Rails, Linux, Free/OpenBSD, Mac OS X, PC (95/98/ME & NT/2000/XP/Vista), Sparc, VAX/VMS, and other Unices, Vignette CMS, iPlanet, IIS, Apache, mod_rails, Tomcat, Mongrel, Lighty, Samba, NFS, TCP/IP, HTTP, HTTPS, SSL, and OOD</p> <p><i>Programs:</i> TextMate, Most Mac Software, git, SVN, Capistrano, Eclipse, MyEclipse, Microsoft Dev Studio (VC++, VJ++), All MS Office Apps, All Corel Office Apps, Most Adobe Apps, Most Macromedia Apps, Flash, DreamWeaver, FrontPage, NetObjects Fusion, Rose, Visio, ERwin, LaTeX, Emacs/Xemacs, VI, VPN, MySQL/mSQL, Qmail, BIND, JRun, Café/Visual Café, Java Workshop, JBuilder, Forté and NetBeans</p> <p>Familiarity With:</p> <p><i>Languages:</i> J2EE, Portals/Portlets, Flash/Shockwave, ActiveX/ASP/.NET, Tcl/Tk, XML, iCalendar/.ICS, vCard, SmallTalk, Prolog, Lisp, Ada, Fortran, Scheme, Assembly, Cobol, Logo, Tex, Basic and UML</p> <p><i>Architectures:</i> Operating System Level Programming on Mac, PC, Sparc, HP-UX, and Linux/BSD</p> <p><i>Programs:</i> Oracle, DB2, Access, PostgreSQL, ODBC/JDBC, CVS/Subversion, VSS, FileMaker, MS Project, Ghostview/script and Clarisworks</p> <p>Personal:</p> <p>I am an active person who enjoys new challenges. I am able to learn new skills very quickly and easily. The majority of the skills listed above were self-taught. I work well with others as well as on my own. I have had experience working with small to medium-sized teams as well as leading teams. I particularly enjoy taking on new projects and seeing them through to completion.</p>

Employment

December 2005 - Present: Sharp Internet Solutions – *Web Design & Custom Web Application Development*

Continued full-time expansion of web services company, with greater focus on custom web application development and deployment. Significant focus on Ruby on Rails technologies for complete, end-to-end database-backed web applications. Created two start-up companies using Rails technology: **Xprove, Inc.**: www.xprove.com and **adsOOH, LLC**: www.adsooh.com. Expanded repertoire of internet technologies for web site development.

September 2003 – November 2005: Novell – *Lead Technical Engineer*

Worked across multiple departments on multiple large-scale projects. Built solutions using industry-standard tools, such as Java Servlets and Portals. Provided support and ongoing development. Project details include: **NVT (Novell Virtual Terminal)** – Worked as Lead Architect and Developer to design, implement and deploy a single web-based framework and solution for processing online financial transactions. Successfully deployed and utilized across multiple divisions within the company, reducing redundancy and saving costs. **Ximian to Novell Transition** – Worked to facilitate the transition of web tools and content from Ximian to Novell. Provided support for the transition of the Ximian online store to Novell. Helped redesign and repurpose Ximian content and tools for Novell's web sites.

My Account – Worked to architect and develop a single-view solution for customers, partners and employees. Designed a web-based interface for collating and managing all customer information, tools, and services within a single Portal view. Implemented first-stage solution using Java Portal & Portlet solutions.

Content Management: Translation Toolkit – Built a Java toolkit for managing automated content translations via the Vignette content management system and a 3rd-party translation service.

July 2002 – September 2003: Ximian – *eCommerce Engineer and Web Application Programmer*

Worked for small start-up Linux product company, providing eCommerce solution management and Web development and support.

eCommerce "The Store" – Managed and enhanced custom-built eCommerce package for selling software products, support licenses and promotional materials. Tools utilized: PHP and Perl.

Internal and External Web Sites – Provided primary application development and support for internal and external (customer-facing) web sites. Developed application and delivery framework based upon Smarty, a PHP-templating system. Provided support for general web development issues and fixes.

April 2001 - Present: Sharp Designs & Sharp Hosting – *Web & Internet Design, Programming and Hosting*

Continued full-time web design, custom internet-based programming, application/database development via self-started company. Increased web hosting support, including Linux administration. Added custom networking (wired and wireless). Incorporated custom programming and software development. Added hardware sales and support. Built full-capability e-commerce solutions, both commercial and custom-built. Extensive development via PHP and MySQL in a variety of web-based application solutions.

July 2000 – April 2001: Scient – *Lead Developer and Applications Architect*

Worked with Fortune 500 companies architecting web solutions for financial, retail, and healthcare-related projects. Designed database solutions, application architectures and web wireframes for companies such as State Street, Chase Manhattan, Genzyme, and Buy.com. Worked as mentor for new employees in areas of programming and application architecture. Project details include:

Genzyme: eBusiness Assessment Project – Worked as Applications Architect and technology representative investigating current systems and processes, evaluating eBusiness readiness, and recommending changes and processes toward implementing eBusiness solutions.

Buy.com: Backend Redesign Project – Worked as Applications Architect to investigate backend order processing and management systems and processes. Interviewed and analyzed multiple top-tier vendors. Constructed a detailed Vendor Capabilities Matrix and cross-referenced it against gaps and opportunities for the client. Provided a detailed technical analysis/recommendation for re-architecting the backend systems.

State Street: Internet Initiative – Worked as acting Applications Architect and Web Developer helping the client to investigate and design a new customer-centric Internet Portal. Provided technical expertise, database analysis and security assessment throughout the initiative.

Chase: New Online Payments Initiative – Worked as Team Lead in developing an online payment system for Chase. Provided mentoring and support to a small team of J2EE developers. Managed installation and

configuration of custom reporting solution.

July 1999 – June 2000: The MITRE Corporation – Corporate Intranet/Extranet Development & Support

Supported internal application suite as part of corporate Intranet. Built custom Extranet utilizing Java Servlets, Perl, LDAP, HTML/XML & X.509 PKI. Extranet functionality included customized auto-registration, dynamic administration toolkit, collaboration, and document sharing. Leveraged multiple commercial products, including Netscape's iPlanet and Netegrity's SiteMinder. Initial support of in-house workflow application. Provided Java/Perl/Linux mentoring to other staff/co-ops.

January 1998 – Present: Sharp Designs – Web Site Design & Maintenance

Started company to design and develop web sites for individuals, small businesses and large businesses. Focused efforts on 3rd generation design/layout, e-commerce, advanced CGI & multimedia development.

June 1997 - February 1999: Concept Five Technologies, Inc. – Security in Distributed Systems

Lead development of Java version of CORBA Security architecture and implementation (CORBA Sec Level 2 Compliant). Created GUI interfaces, back-end services/systems, and extensions to existing ORBs.

August 1996 – June 1997: Concept Five Technologies, Inc. - Collaborative Technologies Directorate

Developed Client/Server applications to support real-time collaboration using Java and CORBA. Also developed online web services and provided general web consulting and training.

January 1996 - August 1996: Mitretek Systems, Inc. - Collaborative Technologies

Developed initial prototype of Java-based real-time collaborative system. Created and maintained online training and documentation services. Served as POC for Java and HTML questions.

July 1995 - January 1996: The MITRE Corporation - Information Security Division

Worked on porting digital signature code to laptop/SmartCard architecture. Built custom online services, such as a personalized home-page maker and a company-wide online survey. Provided UNIX system administration for Division server. Provided Web Master services for Division server. Served as Department POC for HTML questions and training. Built Java-based real-time financial trading application.

Activities

Current:

I enjoy a number of activities and hobbies, particularly related to computers and the outdoors. These include biking, hiking, camping, skiing/snowboarding, rollerblading, skeet shooting, hunting and most sports. I also enjoy learning alternate OSes (Linux, BSDs, etc.), building home networks, learning new computer skills/languages/architectures, and being the family "computer guy." And, of course, my newest "hobby": my growing family including my son, Caleb and my daughter, Emma.

Amherst College:

Computer Center Supervisor, Teaching Assistant in Beginning Programming Course, Math and Computer Science Tutor, Varsity Ski Team, Fencing Club, Karate Club, Ultimate Frisbee Club, A Cappella Singing Group, WAMH DJ (College Radio Station), InterVarsity Christian Fellowship.

Steamboat Springs High School:

National Honor Society, Speech and Debate Team, Student Council, Varsity Football (All Conference Center), Spanish Club (President), International Relations Club, Junior Engineering Technical Society (President), Youth Group, Jazz Band, Marching Band.

Ruby on Rails Projects

Dan Sharp – Sharp Internet Solutions

(970) 672-0337 (phone)
dan@sharpinternetsolutions.com

(206) 337-1398 (fax)
www.sharpinternetsolutions.com

Purpose

This document highlights the Ruby on Rails projects that I have worked on over the past three years. The goal is to describe in sufficient detail the projects themselves and the technologies used in order to showcase my experience in Ruby on Rails and depth and breadth of my web application development expertise.

Breadth of Experience

I have worked across all aspects of the Rails architecture for the past 3+ years. I have used and developed in Rails on Windows, Linux boxes and Macs and am comfortable with any of these three platforms for Rails development. I prefer OS X plus TextMate but am comfortable on all three platforms.

Most of the projects were individual undertakings, but I have also team-programmed on more than one project. I have used SVN and Git for source control and Capistrano for deployment. I have set up and managed deployment solutions both as stand-alone dedicated servers and via shared hosting services in combination with my web hosting company: Sharp Hosting. I have installed, configured and managed Apache + mod_rails, Lighty and Mongrel clusters.

I have kept up with current Rails and Rails-related developments. I tend to stick to the latest stable release, primarily because I focus on project work using stable, deployable technologies. Thus, I do not live on Edge Rails and only adopt beta/edge technology/plugins when necessary.

I am not adverse to modifying or monkey-patching source code, either in Rails or in plugins, when necessary. Some patches were done in older Rails versions to support features before they became available. I have enough software architecture experience from previous employment that I am able to see and build “the big picture” just as well as diving into the nitty-gritty code. I enjoy both aspects of software development: architecture/design and development/implementation.

I also have experience taking an idea and building a web-based business around it. Two of the projects developed into companies (one S-corp and one LLC). These involved partners and business decisions and aspects beyond simple code development. I am comfortable working with marketing, business and sales people in regards to technology projects/solutions. I work well on my own or on a team. I am not a manager, however. I like to “get my hands dirty”.

The one area, I admit, where my experience is less than desired is in the area of testing. All of these projects have undergone extensive testing, but usually “live-style” testing, meaning: trying everything out in the browser. I have written some unit and functional tests and have started to learn some of the testing frameworks and ideas such as TDD/BDD. I am definitely for testing and hope/plan to learn more and apply more in this area for future projects and Rails ventures.

Projects

The projects I have been involved with that utilized Ruby on Rails technologies are listed here. Greater detail for each project can be found below.

- **adsOOH LLC** – a digital signage / Out-Of-Home marketing services company
- **Xprove, Inc.** – a post-production review and approve services company
- **The FORC** – a community site for Regulatory Insurance law
- **The Peterson Page** – a members-only job posting community
- **RV Focus Law** – a document management site for the RV industry
- **TookABook** – a book sharing service
- **Cheers!** – a helps/service management project

Access

There are varying levels of demo/live access to these sites. Some have public-facing pages. Some require administrative access. Some have staging/demo servers. Some have screencasts. And some never launched. If you are interested in further details from any of these projects, let me know and we can arrange access or code samples.

adsOOH LLC – a digital signage / Out-Of-Home marketing services company

www.ads00h.com

adsOOH LLC

Project/Company Description

adsOOH LLC is a web-based middle-ware company conceived in early 2008 to offer services to the digital Out-Of-Home (OOH) industry. In short, adsOOH allows digital signage networks to sell their advertising “inventory” (on-screen ad-space + run-time slots) to local businesses, an opportunity previously ignored as too cost prohibitive. adsOOH provides a branded web portal for selling and managing video commercial ad buys. It involves an extensive amount of programming and leverages many of the latest Rails technologies.

adsOOH is a three-tier system, offering unique and secured access to (1) networks, for managing their inventory and sales, (2) producers, for managing video production and sales, and (3) customers, for managing their ad purchases and video deliveries. Customers access the system via branded, subdomain-managed “portals”, one for each digital signage network.

A full-featured demo site showcasing all three tiers has been developed. These run as live transactions except for (a) no actual payments are made... they are passed through as test transactions, and (b) no actual emails are sent... they are stored and displayed via a custom email “previewer” mechanism.

Technologies Used

Built and deployed using the latest Rails at the time and recently upgraded to Rails 2.2.2 (with 2.3.x upgrade coming very soon). Source control is handled via SVN originally but upgraded in mid-2008 to Git using a remote Git repository. Deployment via Capistrano 2.x to shared hosting server running Apache 2.x and mod_rails + MySQL 5.0.x. Developed using TextMate on OS X and collaborated via Skype. Some Rails and supporting technologies leveraged for this project:

- Rails 2.2.2 and beyond
- MySQL 5.0.x
- Apache 2.x + mod_rails on shared hosting (for demo site)
- SVN + Git for source control
- Capistrano 2.x for deployment
- Screencasts created via SnapzPro & IShowU and QuickTime Pro
- PHP uploader + mod_rewrite for large-file uploading with status
- Active Scaffold integration and customization for administrative toolkit
- Active Merchant integration and customization via BrainTree for Merchant Processing
- User management via restful_authentication + role_requirement
- Subdomain management for fixed + custom subdomains via subdomain-fu
- Video conversion to flash video via spawn + ffmpeg conversion script
- SSL support for payment pages via ssl_requirement
- Google Maps integration with custom JavaScript + JSON for custom on-the-fly maps
- State machine via AASM gem
- Dynamic filtering mechanism – networks can create filterable categories on-the-fly
- Unit + Functional tests + custom bootstrap system
- Saas, Haml and jQuery used in some places
- Custom form builders
- Extensive use of Partial and Subtemplates across all tiers
- Quad-tiered namespaced routes: Network, Producer, Customer, Administrator

Challenges/Innovations

The adsOOH system has, by far, been the most complex and extensive Rails project I’ve undertaken to date. It has it all: multiple tiers of access, user management, financial transactions (via Active Merchant), Google Maps integration, Ajax, video upload + conversion, jQuery, Saas + Haml, messaging, file attachments, bank transactions, credit card transactions, reporting, process flow management via state machine, user levels + roles, dynamic filters, searching, sorting, pagination, and more. Nearly 7,500 lines of code across 120 classes!

Xprove, Inc. – a post-production review and approve services company

www.xprove.com

Xprove, Inc.

Project Description

Xprove, Inc. is a web-based services company launched in 2006 to provide online video review and approve services to post-production companies. It was the first Rails project undertaken and began prior to Rails 1.0. Xprove was built as a team of three using shared programming practices across diverse geographies. The application was launched on a dedicated server and continues to service hundreds of clients to this day.

Technologies Used

Built and deployed initially using Rails 1.0 and later 1.2.x. Developed using TextMate on OS X and collaborated via SVN and Skype. Some Rails and supporting technologies leveraged for this project:

- Rails 1.2.x
- MySQL 4.1.x
- Lighttpd (Lighty) deployment
- Web uploader integration for large-file uploading
- SVN for source control
- Capistrano 1.x for deployment
- Prototype
- Custom Merchant API integration for Credit Card support (reference storage only)
- Custom Mime Type support
- Layered layouts
- Custom cron scripts
- Ruby shell scrips for administrative tasks

Challenges/Innovations

Xprove was built before and as Rails as a framework was just beginning to emerge into the “public eye”. There were a number of implementation challenges due to the newness of the technology. We built dynamic subdomain support before Rails officially included it using custom route management and filters. We integrated a PHP-based uploader with hand-shake integration between Lighty and Apache. We created and integrated custom mime-type support before Rails fully supported mime-types. We built a custom merchant API to interface in real-time with our merchant account provider, offering credit card purchases and reference-based account storage (no CC numbers stored).

The FORC – a community site for Regulatory Insurance law

www.forc.org

The Federation of Regulatory Counsel, Inc.

Project Description

The Federation of Regulatory Counsel, Inc. (FORC) hired me to take their existing Joomla!-based site and completely rebuild it to support a much greater set of document management features. The first pass was completed in 2007 using early Rails technology. In late 2008-early 2009, the FORC again contracted me to upgrade a number of features of the site. This description includes the latest round of development.

The FORC site provides public access to Journals, which include multiple articles by multiple authors, Alerts, which include short “blurbs” organized by state/region, Events, such as upcoming board meetings or special seminars, and Member search/display using a graphical map of the US + member lists. People can sign up for various mailing lists to be able to receive the journals, alerts and event notices via email.

The internal management portion of the website is called “The Toolkit”. This provides access for FORC members to create, manage and publish articles, journals, alerts and events. There is a member management component whereby members with appropriate access rights can manage the member list. Recently, a discussion forum based off of Beast was integrated for member discussions and all-member notices. The email system was improved to include HTML-styled emails. The Events system was upgraded to include file attachments and iCal (.ICS) integration.

Technologies Used

Built and deployed using Rails 1.0.x originally and redeployed using Rails 2.2.2. Developed using TextMate on OS X. Source control managed via git and remote git repository. Deployed using Capistrano 2.x to a shared staging Apache server with mod_rails. Some Rails and supporting technologies leveraged for this project:

- Rails 2.2.2
- MySQL 5.0.x
- Apache 2.x + mod_rails
- Capistrano 2.x
- Prototype + jQuery + Ajax
- Custom modifications of ar_mailer + spawn for large-volume email distribution
- File attachments via attachment_fu
- Graphical calendar implementation via calendar_date_select
- ICS/iCal implementation via icalendar plugin
- Custom Paginator implementation (this was pre-will_paginate)
- User authentication via custom authentication (this was pre-acts_as_authenticated)
- TinyMCE integration for “fancy text areas”
- Email notification of signups
- Uploaded file management & storage
- Beast integration and customization with tiered access levels
- Custom TinyMCE alteration to support in-line references converted to end-notes
- Complex access control and user levels
- Dynamic list creation, alteration and ordering
- Auto-PDF generation using HTMLDOC and custom layouts/mime types

Challenges/Innovations

This was a site that involved retrofitting technology improvements around an existing visual design. The conversion of Joomla! pages to Rails-delivered pages involved some headaches and challenges. This project involved a lot of new and interesting Rails technologies, each with their own challenges. The two biggest challenges were: (1) building a TinyMCE plugin to support end-note references. Authors could select text in an article, flag it as an end-note via the TinyMCE button, and upon publishing of the article, the reference text would be stripped and auto-organized as end-notes. Various people have access to the publish functionality, allowing them to mark Journals, Alerts and Events as “published”, thus making them visible on the public site as well as sending them to the appropriate mailing list. One particular challenge was taking the original project code, built on Rails 1.0.x and upgrading it to Rails 2.2.2 throughout the codebase.

The Peterson Page – a members-only job posting community

www.thepetersonpage.com

The Peterson Page

Project Description

The law firm Baker & Daniels LLP desired to create a members-only community for former members of the administration of Mayor Bart Peterson. The design was retrofitted from Baker & Daniels site design templates. The site provides user profiles, user listing, job postings, admin features and multi-format views, including PDF, Spreadsheet (CSV), and full-vs-compact listings. Future expansion plans include event/calendar management, forums, custom email templates, and more. The admin features allow for management of user profiles and job postings by admin-flagged users.

Technologies Used

Built and deployed using Rails 2.0.2. Developed using TextMate on OS X. Source control managed via Git and deployed using Capistrano 2.0 to a shared Apache server running mod_rails at Sharp Hosting. Some Rails and supporting technologies leveraged for this project:

- Rails 2.0.2
- MySQL 5.0.x
- Apache 2.x + mod_rails
- Capistrano 2.x
- Git + remote git server
- PDF auto-generation via RailsPDF
- CSV auto-generation via FasterCSV
- Pagination via will_paginate
- Custom formats for PDF, CSV, and “full list” displays
- Ferret and acts_as_ferret for search functionality
- User authentication via modified restful_authentication plugin
- Email via Observers
- Custom testing with data loader for bootstrapping data
- TinyMCE integration for “fancy text areas”

Challenges/Innovations

The existing design had some challenges in that it leveraged some very convoluted CSS styling and did not have good multi-browser/multi-platform support. The other challenge was getting PDF and CSV generation to work well. I chose RailsPDF for PDF generation and FasterCSV for CSV generation.

RV Focus Law – a document management site for the RV industry

Project Description

The law firm Baker & Daniels LLP hired me to upgrade their document management site for the RV industry: www.rvfocustlaw.com. This site offers a searchable/filterable list of documents to members. The prior site had been implemented using Joomla! but was too cumbersome for people to use effectively. I proposed rebuilding the site in Rails and mapping a design similar to the design (at the time) of the Baker & Daniels website. The RV document management site never launched due to internal management issues at Baker & Daniels. However, the project was completed and the site remains working to this day on my staging server.

This project involved a heavy amount of Prototype and custom JavaScript/Ajax work. The searching, filtering, and displaying of documents, organized by category and state, is handled via Ajax as much as possible. A user can multi-select categories and/or states to select documents that match filter criteria. Users can sign up to request access to the system. Administrators can manage pending user accounts and approve or deny them access. Documents can be inserted into the system either as uploaded documents or using TinyMCE to enter document content.

Technologies Used

Built and deployed using Rails 2.0.2. Developed using TextMate on OS X. Source control managed via SVN and shared across 2-person team. Deployed using Capistrano 1.x to a shared staging Apache server calling FastCGI at Sharp Hosting. Some Rails and supporting technologies leveraged for this project:

- Rails 2.0.2
- MySQL 4.1.x
- Apache 1.3.x + FastCGI
- Capistrano 1.x
- Prototype + Ajax
- Nested Layouts plugin
- Custom Paginator implementation (this was pre-will_paginate)
- User authentication via acts_as_authenticated
- TinyMCE integration for “fancy text areas”
- Email notification of signups
- Uploaded file management & storage

Challenges/Innovations

As with The Peterson Page, there were some challenges in using an existing design. Furthermore, the user interface for selecting states and/or categories required extensive Ajax work and testing to make it work right. We implemented the system using three user classes: users, admins and superadmins. The users could browse documents. The admins could manage documents and manage user accounts. The superadmins could convert a user to an admin, as well as delete/modify admin accounts. At the time, the best user management plugin was “acts_as_authenticated”. We used this plugin, with some modifications.

TookABook – a book sharing service

www.tookabook.com

TookABook

Project Description

TookABook is an idea spawned from some online forum discussions regarding book sharing. In a particular forum that I frequent, there were some threads where people suggested the possibility of posting books they had and people in the forum could request them. There was a lot of enthusiasm for the concept but no real sense of how it would work.

So I thought about a web-service for book sharing. Combining social-networking with book distribution and a very modest profit model. Basically, the system would be branded for each “community” and within that community, members could post book lists and wish lists. When a match (book-to-wish-list) is made, the requestor pays a modest fee to cover shipping costs and a very small commission (to TookABook). The owner of the book then ships the book (or books) to the requestor. Upon receipt, the money is credited to the owner to reimburse them for shipping costs.

Effectively, this becomes a business model not unlike Amazon’s used book sellers except (a) it’s within tighter communities and (b) it’s not profit-based for the book owners... it’s sharing-based. It’s the geographically-disperse version of letting your friends browse your bookshelves and say “that one looks interesting... can I borrow that?”

To date, the service has not launched, primarily because my time was taken with other ventures and the development is not complete. However, I still have high hopes for this service, in spite of a few other similar offerings already out there.

Technologies Used

Built and deployed using Rails 2.0.x. Developed using TextMate on OS X. Source control managed manually. Some Rails and supporting technologies leveraged for this project:

- Rails 2.0.x
- MySQL 5.0.x
- Prototype + Ajax
- Subdomain support for “communities”
- User management via restful_authentication
- Transition states handled via AASM
- Amazon/ECS integration

Challenges/Innovations

Most of the challenge of this project revolved around modeling the interactions between book owners and book requestors. For example, how should the system handle the transition of the book from requested, to request acknowledged to book sent to book received and such. And when should the financial transactions take place? I also spent some time integrating the Amazon/ECS API for ASIN lookups. This way, book information is immediately accessible simply by entering the title and author or the ISBN number.

Cheers! – a helps/service management project

(no website)

Cheers!

Project Description

Cheers! Is another idea for a web-based service that I began to develop. My wife started a service ministry within our church whereby she gathered people willing to help in a variety of skills/tasks and paired them with people in the church or community in need of help. As the ministry took off, it became more difficult to manage the data set of “helpers” with “helpees”. What if there were a web-based application to handle the data management and the needs-pairing? Thus was born the Cheers! Project. As with TookABook, this is a project that received significant development interest but has yet to be completed and launched.

The system I built started as a deep customization of Active Scaffold. I learned all the ins-and-outs of the AS APIs and how to properly set up model interactions. I built a nice administrative interface that allows a “helps manager” enter helpers, enter helps requests and match helpers to requests based upon dynamic categories.

Ultimately, the vision of the project is that this system could be portal-customized for non-profit organizations and used to manage the full spectrum of a “helps” ministry, including monetary donations. Using Active Scaffold as a starting point allowed me to quickly figure out the objects and interactions within the system.

To date, the service has not launched, primarily because my time was taken with other ventures and the development is not complete. As with TookABook, this idea still has great potential, perhaps as an ASP model of a branded portal-like service offering to churches and other service-oriented organizations.

Technologies Used

Built and deployed using Rails 2.0.x. Developed using TextMate on OS X. Source control managed manually. Some Rails and supporting technologies leveraged for this project:

- Rails 2.0.x
- MySQL 5.0.x
- ActiveSupport
- Prototype + Ajax
- Subdomain support for “portals”
- User management via restful_authentication
- ActiveCalendar + jsCalendar

Challenges/Innovations

By far, the biggest challenge was understanding ActiveSupport. It is a very robust technology for model interaction and “administrative-style” UIs. The API at the time was in active development and not as well-documented as I would have liked. Thus, there was a lot of source browsing, forum discussions and trail-and-error to get AS working as I envisioned. AS is also very Ajax-heavy. I found a number of tools to debug Ajax/JavaScript issues.