

Web Application Technologies

Surveying The Landscape

Gunther Birznieks

gunther@eXtropia.com

Overview

- Open Discussion
 - Feel Free to Ask Questions Anytime...
- Covers *dynamic* application technologies
 - Perl, Java, PHP
 - ColdFusion, ASP (if we have time)
- Features and Pros/Issues
- Free sites
- Emphasis – “*Open Source*” solutions. **This NOT a Vendor/Language-specific Talk!**
- Technology Disclaimer: eXtropia is a free software firm (Perl & Java)

Technologies

- The Stereotypes
 - You may not agree – But they exist
 - Perl hackers
 - Java corporate
 - PHP for small, easy web sites
- Obviously, Stereotypes are wrong...
 - But there is a grain of truth
 - Languages are designed for certain tasks that they excel at which involves tradeoffs

Language Dissection

- We're not here to deride any languages, just to explain them.
- The slides here are meant to present a fairly even way of presenting the technologies based on our real-world experiences
 - Opinion Alert: I do believe in using the right tool for the right job *and to suit the environment/infrastructure*
 - eg The right tool for the right job may not be the right tool if your environment uses another tool for everything else.

Language Dissection

- Guidelines of why some languages may be better suited for some environments than others...
- Rather than focusing on the cons, we'll try our best to focus on the pros of each language
 - *Purpose: to give you a guideline to aid your own decision making and to help raise questions in your mind to ask those speakers about their various solutions when you attend language-specific talks here.*

Technology vs Languages

- Technology – based on a language but not necessarily equivalent
- Java
 - JSPs and Servlets
- Perl
 - Traditionally known for CGI
 - But also consider mod_perl, FastCGI, AxKit, EmbPerl, CGI::Speedy, Apache::ASP, etc.
- PHP
 - For the most part, the language is the technology

Java Technology

- Origin: Sun
- JSPs and Servlets
- Java Hype (aka Powerful Marketing)
 - For some good reasons, some not
- Definitions - some of which you may disagree...
 - Java – A language designed to run on any platform a JVM (Java Virtual Machine) exists
 - Servlets – A technology allowing Java to run inside a web server dynamically
 - JSPs (Java Server Pages) – A technology to allow Java to be embedded in HTML pages

Java Technology

- As a rule of thumb, the pros of Java Servlet technology include:
 - The applications are cached on the web server and may run many times (unlike CGI)
 - The data for the app may also be cached (eg database connection pooling)
 - Live Objects may be shared across sessions and application contexts (not just requests)
 - The JDK is a rich standardized set of technology
 - After years of Sun funding, the JDK has many classes to do just about every major task

Java *Language* Features

- Of course, there are many features but I list the ones that I feel are most relevant to web apps
- Backed by Sun
 - I know its odd to list this as a feature, but some of its success is certainly due to huge \$ behind R&D especially enterprise R&D (j2ee)
- Intermediate Compiled language
- Cross Platform
- Built-In multithreading

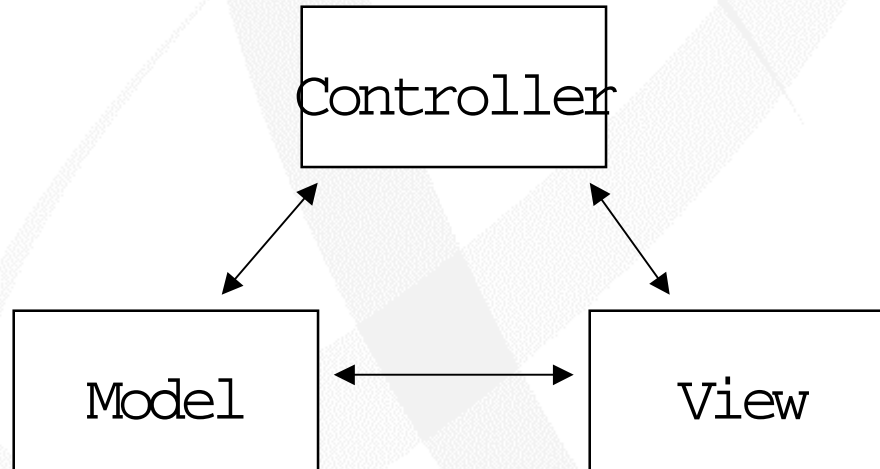
More Java Pros

- Servlet engines exist for every major webserver.
 - You can develop on Apache and sell to a Netscape or IIS/NT user
- JSPs give a standard form of *templates*
 - Templates allow HTML designers closer access to the application – saves developer time
- JSPs are compiled into servlets so share the same benefits

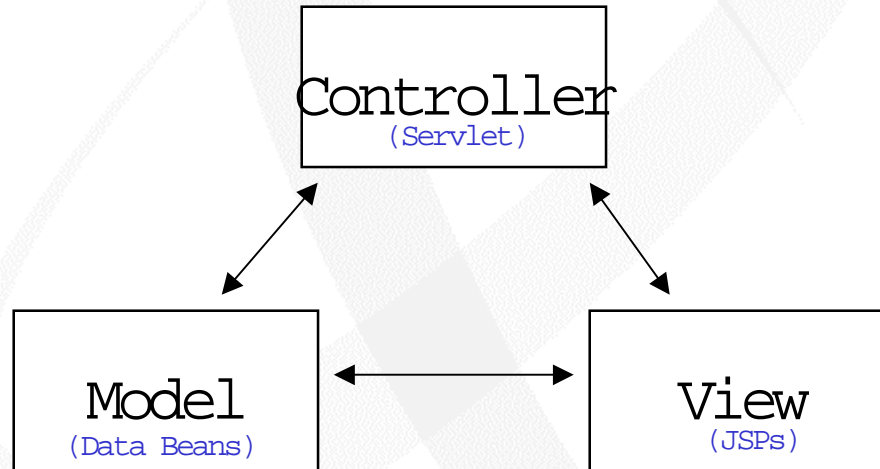
More Java Pros

- JSPs support a special mode called dispatching
- Dispatching allows you to write all the control code in a servlet and then use the JSPs just for views.
 - This is a very powerful model
 - Allows JSPs to just contain HTML
 - Yet allow a servlet and accompanying support classes to control the application workflow and access to the data model
 - Model-View-Controller (eg Struts, Turbine)

Model-View-Controller



Model-View-Controller



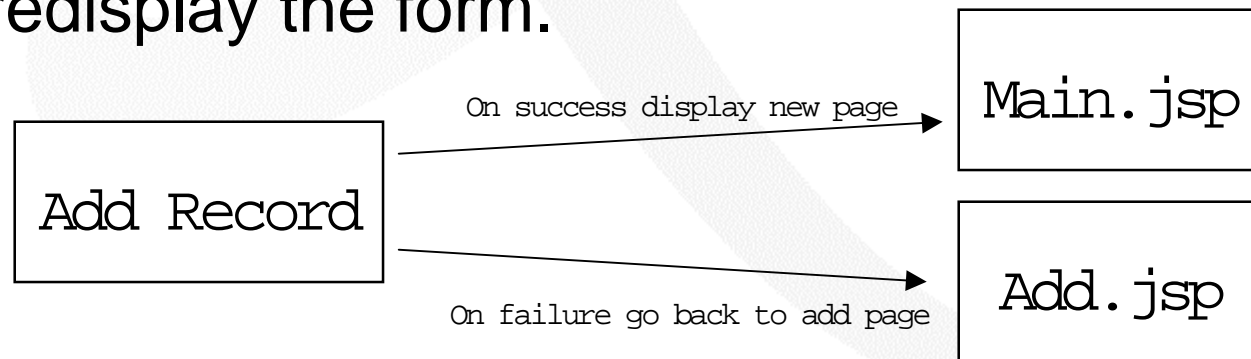
MVC Example

- Request comes in for adding a record
- Controller Servlet determines from user state (ie gesture pattern) what the user wanted to do (add a record)
 - For example, perhaps submit button named `action_addRecord` was clicked.

```
<INPUT TYPE="SUBMIT" NAME="action_addRecord" VALUE="Add Record">
```

MVC Example

- Controller then calls Model bean called DatabaseBean and tells it to add the record.
 - Eg `databaseBean.addRecord();`
- If the record add worked, then the main DB view JSP file is called.
- If the record add failed (eg bad data input), then the add form JSP file is called again to redisplay the form.



More Java Pros

- Security
 - Java as a language supports the ability to specify security policies for the virtual machine
 - eg no files to be read or written from the server.
 - The concept is known as a *sandbox*.
 - A less known fact is that Perl also has this concept, but approaches the Web security problem from a different perspective discussed later...

Java Issues

- Class reloading
 - Java is object-oriented, so class loading is a natural extension of setting up a web application
 - Unfortunately, the way JVMs were implemented, it is a non-trivial problem to reload a class in an active servlet without restarting the entire engine.

Java Issues

- Debugging Support
 - Still in its infancy
 - To be fair, debugging on the web is an oft-overlooked feature for languages
 - The problem is that JSPs are supposed to be easy but when things are all wrapped inside of JSP exceptions you can lose valuable debug data (especially if using servlet->JSP dispatching)
 - Usually easier with a commercial IDE (eg Internet Edition of Forte)

Java Issues

- Verbosity
 - Java, being strongly typed, is a verbose language. Downcasting is quite common.
 - eg `String name = (String)map.get("name");`
 - versus Perl 's
 - `$name = $map{"name"}`

Java Issues

- Verbosity
 - Some argue that the verbosity increases the number of lines of code which can increase errors....
 - Others would argue that the verbosity is mostly strong typing which catches errors at compile time (and thus are easier bugs to fix)

Java Issues

- Lack of a *Large* Open Source Community
 - OK, there is clearly a community with Jakarta-Tomcat, Cocoon, Xerces, SOAP and many other projects BUT...
 - Try and find a free web application written in Java Servlets? At most you usually find about 1-2 of the flavor you are looking for.
 - Perl and PHP literally have at least a hundred or more easy to download and free web apps to do just about everything as a template.
 - Few people in Java community open source their apps. It's common for a company whose employee develops a Java app to see \$ signs and start to sell the app.

Free Java Web Sites

- The original
 - Gamelan (<http://www.gamelan.com/>)
 - Owned by EarthWeb, General Java Resource
- Internet.com's JavaBoutique (<http://www.javaboutique.com/>)
- eXtropia – (<http://www.extropia.com/development/java/>)
 - WebDB, WebCal (Calendar/Tasks)
- CoolServlets (<http://www.coolservlets.com/>)
 - Forum, Mailing List
- JavaCorporate (<http://www.javacorporate.com/>)
 - Forum, Search engine (DB)
- Meta-Directories
 - <http://www.servletsource.com/software.jsp>

Perl Technology

- Origins: Sysadmin Language (by Larry Wall)
- Grew quickly in its feature set esp. text parsing
- Popularly synonymous with CGI
 - Common Gateway Interface
- Perl was one of the first Web languages
 - We started Selena Sol's Scripts Archive on Perl 7 years ago
 - Many free resources (although varying quality)
- Also includes mod_perl, CGI::Speedy, other Perl acceleration solutions

Perl Pros

- It's been around since the beginning of the Web
- Ubiquitous at ISPs
 - Almost can't go wrong writing a CGI for cross-ISP compatibility
 - CGI is slow though (but may be fast enough for many website needs)
- Can enhance power of CGI by using an accelerator (eg `mod_perl`, `CGI::Speedy`, Binary Evolution's `Velocigen`, ActiveState's `PerlEx`)

Perl Pros

- Note: mod_perl is more than a Perl accelerator
 - It also interfaces with Apache internals
 - This is VERY powerful and allows you to do many things you cannot do with Java
 - Java Servlet/JSP is a standard that works across web servers
 - Mod_perl is for Apache
- Another pro is that Perl has a strong open source community in terms of quantity of solutions available – partially due to its age and being so open from the beginning

Perl Pros

- CPAN – Comprehensive Perl Archive Network
 - A rich set of open source Perl libraries
 - Compare JavaSoft's JDK
 - Similar in the number of classes
 - Dissimilar in that
 - CPAN is fully open source and a cooperation of many developers adding onto it
 - This is it's strength because there are so many swiss army knives on CPAN

Perl Pros

- Template Engine Choice
 - There are at least 10 different very powerful HTML template engines for Perl
 - Each written for different purposes
 - Some for managing large sites (content management)
 - Some for just self-contained web application HTML templating
 - Java by contrast has JSPs as a defacto standard (even if there are alternatives). While it's nice to have a standard, choice is also a Pro

Perl Pros

- Free Application Choices
 - Perl apps have been so ubiquitous for over the last seven years on the Web, that there are many many open source solutions solving the same problem in slightly different ways.
 - Many different BBSes, web stores, form processors, web db front-ends etc.
 - Just pick and choose from simple to complex

Perl Pros

- Power App Case Study: AxKit
 - AxKit is an XML Template/Portal processing engine written for mod_perl
 - Cocoon is the similar project for Java
 - For serving previously cached pages, AxKit, because it hooks into Apache internals can hand control back into Apache and tell it to serve the file
 - A cross-platform servlet such as Cocoon can run on platforms other than Apache but lack the integration with the internals for eeking out more speed/power

Perl Pros

- Security
 - Perl having been written for sysadmins originally has a special mode called *taintmode*
 - Puts Perl in a sort of paranoid secure watchdog mode
 - Essentially allows you to have the Perl believe that all user input is “tainted” and therefore cannot be used to perform operations such as file manipulation unless the programmer has specifically “untainted” the data manually.

Perl Issues

- CGI is slow
- Mod_perl and other persistent engines can solve this problem
 - Yet Mod_perl is not as easy to code
 - You have to be more disciplined because of cached data and packages
 - Few solutions exist between these extremes and offered at ISPs on a wide-scale basis
 - Projects to improve this include:
 - CGI::Speedy – A very easy way to speed up CGI scripts
 - Mod_perl 2.0 (has many improvements to make user and ISP life easier)

Perl Issues

- Perl is not multi-threaded
 - This means that data structures that need to be shared among sessions need to be persisted to disk or an IPC/shared memory mechanism
 - A Perl based application communicating with a back-end cannot also operate using call-backs because the back-end won't know which Perl process initiated the request
- Note, this is not an issue for most web apps.

Free Perl (CGI) Sites

- The originals (Mostly Perl based)
 - eXtropia (<http://www.extropia.com/>)
 - Originally known as Selena Sol's Scripts Archive
 - Founded as an offshoot from work done at the EFF
 - Electronic Frontier Foundation (<http://www.eff.org/>)
 - Currently in active development
 - Matt Wright's Scripts Archive
 - <http://www.worldwidemart.com/scripts/>
 - Older scripts, but very simple which can be excellent for beginners
 - Meta-Site Indexes for CGI
 - <http://www.cgi-resources.com/>

PHP Technology

- Origin: Designed for the Web
- This makes PHP very different from Java and Perl
 - Essentially a powerful template language
- From the PHP FAQ on the *php.net* site:
 - “The biggest advantage of PHP over Perl is that PHP was designed for scripting for the web where Perl was designed to do a lot more and can because of this get very complicated. “
 - This is also somewhat true of Java.
 - However, I think the statement “very complicated” is not so accurate. More later...

PHP Pros

- Designed as a scripting language
 - Like Perl, this makes it easy to change a page and test changes immediately
- Designed to be easy
- PHP 4 comes with a bit more of a framework than other languages for doing things natively such as session handling

PHP Pros

- PHP, like Perl, also has a large open source community
 - Although I think Perl is stronger in this respect. Perl's age and capability to be used for more than the web has given it a larger community that is not just web-related.
 - eg A survey of meta indexes indicates more Perl/CGI applications than PHP applications listed...
 - Scriptsearch.com (144 Perl, 17 PHP, 2 Java)
 - Resourceindex.com (2317 Perl, 481 PHP)

PHP Pros

- 95% of what you want to do with the web is basically built-in.
 - Although I mention the framework features, I am also talking about useful web external libraries
 - At most Perl has CGI.pm built into the standard distribution
 - Likewise Java requires J2EE libraries to use Servlets and code JSPs
 - Of course for both Perl and Java, it is relatively easy to download extra libraries but it does take time to figure out what you need and want.

PHP Pros

- For the speed that PHP offers (in-process script handling in Apache), PHP is surprisingly easy to set up for an ISP
 - Servlets and Mod_perl both rely on a shared memory model
 - This makes it extraordinarily hard to support an open shared environment
 - By default the model in PHP is to rerun the script from scratch each time
 - Of course this is also a drawback because it means that speed enhancements using data caching are limited compared with mod_perl and Servlets

PHP Issues

- OK, PHP is a great language for the previous issues... but here's where it starts to fray a bit.
- The advantage of PHP is that the language is simple. But it is also its architectural disadvantage.
 - Note: by now you should be seeing in this talk that most features of every technology discussed here are both their advantage and their disadvantage depending on how you look at it.

PHP Issues

- First, the database access commands as taught to new programmers in the manual are very easy to access a specific database
 - But make it annoying to switch
 - MySQL commands are prefixed with `mysql_` and PostGres commands are prefixed with `pg_`. Changing `mysql_connect()` to `pg_connect()` in pages that use the database is very annoying.
- Contrast with Perl DBI or Java JDBC
- Note: PHPLIB is a project that supports DB independence but it isn't part of the core documentation.

PHP Issues

- Lack of Model-View-Controller architecture
 - To be fair this is a problem with all templates including Java JSPs
 - Also, it is possible to write an app to this architecture using templates – it just becomes *less obvious*.
- Template languages generally teach beginners to think that every HTML page is a program.
 - Bad because HTML pages aren't programs. They are user interfaces.

PHP Issues

- Lack of Model-View-Controller (Continued)
 - This is not important in simple apps like a search query/search results app.
 - It is important in more complex workflow apps such as groupware.
 - `<FORM>` tags that control where the pages go allow the developer to control the workflow. But it breaks down when you have a page with something like an edit and delete button – Do they POST to the same page?

PHP Issues

- Lack of Model-View-Controller (Continued)
 - It's better to have a *controlling* program that is always getting POSTed to and determines based on logic which VIEW (or PHP page) to display
 - Java Servlets and JSPs support this straight out of the box using servlet dispatching. App frameworks such as ours and struts support this.

Free PHP Sites

- Meta-Site Indexes for PHP
 - PHP Resource Index (<http://php.resourceindex.com/>)
 - Sister site to the CGI Resource Index
 - PHP Wizard (<http://www.phpwizard.net/>)
 - PHP Developer (<http://phpdeveloper.org/>)
 - Has a nice Freshmeat Style

Technology Summary

- Points to consider based on what has been discussed:
 - Is my project small or large?
 - Am I part of a larger group using another language?
 - Am I new to programming?
 - Will my project/product need to be sold?
 - Is there a free open source version of what I am trying to accomplish?

Technology Summary

- All the technologies are “good” and “useful”.
 - But in different ways
- The ones you use, and which talks you attend at ApacheCon will probably guide you towards your solution.
- Acknowledgements to Ho Ming Shun, Eric Tachibana, ChinWan Lim

Open Discussion

Questions?