



Building ArsDigita Portals

Frank Bergmann <fbergmann@competitiveness.com>

Barcelona, December 18th, 2000

Content

Introduction

- Who are we and why do I give this lecture?
- What Is a Portal?
- What is ACS?
- ACS Modules
- Online Demos
- What do we do with ACS?

ACS Architecture

- Windows Against Internet Programming
- ACS in its Context
- Java Against ACS Layer Cake
- Other Web Architectures
- Why ACS?

Case Study

- Target Group
- Why Join the Portal?
- Additional Contents
- Which ACS Modules?
- Marketing
- Make Money
- Why didn't I do it myself?

How to Build Your Portal

- Become an ACS developer
- Get an ACS Server
- Build a Simple ACS Portal
- Build Custom Modules
- Form Your Community
- Make Money

1. Introduction

Introduction

- Who are we and why do I give this lecture?
- What is a Portal?
- What is ACS?
- ACS Modules
- Online Demos
- What do we do with ACS?

Who are we and why do I give this lecture?

Who are we?

- The Cluster Competitiveness Group, S.A.
- Located in Barcelona (Cerdanyola)

Before

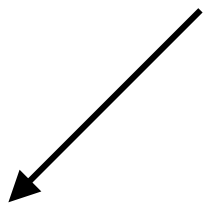
- During 7 year been a strategic consulting company for “Clusters”

Now

- Internet Startup with 60 employees
- Actively developing ACS as our development platform, creating a “Community”

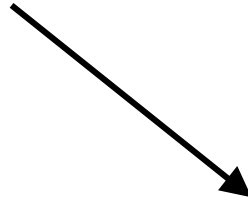
What is Portal?

- Portal == Homepage + Starting Point



Vertical

- Everything about one subject



Horizontal

- General Access

What is ACS?

- It's free
- It's for building portals
- It works out of the box.
- It's a "Toolkit"
 - Library of preconfigured elements
 - Defined way of working
 - Comes with "Philosophy".

What is ACS (2)?

To make an ACS system run you need

- A computer with 128MByte RAM and 6GByte hard disk
- Linux Operating System (Suse 6.4 or Red Hat 7.x)
- AOLServer (not Apache)
- Oracle 8.1.6 Database

and

- You have to configure/customize several modules.

ACS Modules

- [/doc/address-book](#) -- address book
- [/doc/adserver](#) – banner ads
- [/doc/bannerideas](#) – banner ideas
- [/doc/bboard](#) -- the discussion forum system
- [/doc/bookmarks](#) -- save Internet bookmarks
- [/doc/calendar](#) – an online calendar
- [/doc/chat](#) – online chat
- [/doc/classifieds](#) – classified ads: announce your old motorcycle
- [/doc/clickthrough](#) – statistics on ads
- [/doc/contact-manager](#) -- good for sites with an extranet-y flavor
- [/doc/content-sections](#) -- lets administrators add/manage different content sections of the site
- [/doc/contest](#) -- run give-away contests
- [/doc/custom-sections](#) -- allows group administrators add/manage custom sections of the site
- [/doc/curriculum](#)
- [/doc/directory](#) -- letting users find each other
- [/doc/display](#) -- lets administrators set display settings and upload logo of the site
- [/doc/download](#) -- offering versions of software or other stuff to people
- [/doc/dw](#) -- a toolkit for data warehouse-style queries
- [/doc/ecommerce](#)
- [/doc/education](#)
- [/doc/events](#) -- facilitates online registration for events
- [/doc/faq](#)
- [/doc/file-manager](#) -- simple content management tool
- [/doc/file-storage](#) -- users who don't know HTML can collaboratively maintain a set of files on the server
- [/doc/glossary](#) -- the ArsDigita Glossary, for creating an on-site dictionary of specialized terminology.
- [/doc/homepage](#) -- give your users personal homepages on your system
- [/doc/intranet](#) -- run your organization from this toolkit
- [/doc/neighbor](#) -- Neighbor to Neighbor (hard to explain; see example at <http://photo.net/neighbor/>)
- [/doc/news](#)
- [/doc/partner](#) -- site-wide cobranding
- [/doc/poll](#) -- opinion polls
- [/doc/portals](#) -- using only a Web browser, build some portals for different groups within your organization (or the public)
- [/doc/press](#) -- show press coverage about your site or company
- [/doc/pull-down-menus](#) -- create Macintosh-style menu bars
- [/doc/spam](#) -- send email to a group of users
- [/doc/static](#) -- support for collecting comments, links, and download statistics for static files
- [/doc/survey-simple](#) -- ask the users a series of questions
- [/doc/ticket](#) -- a project and bug tracking system
- [/doc/wp](#) -- WimpyPoint, a replacement for desktop bloatware like PowerPoint, allowing you to create and view presentations in your Web browsers

Online Demos

See Separate Online Demos:

- Chat Module
- Classified Ads Module
- User Management

What do we do with ACS?

See Separate Online Demo

- Online Marketplace Wemo
- Community System
- Intranet

2. ACS Architecture

ACS Architecture

- Windows Against Internet Programming
- ACS in its Context
- Java Against ACS Layer Cake
- Other Web Architectures
- Why ACS?

Windows Against Internet Programming

Windows Programming

- Event driven
- Program runs on local machine
- Very fast connection
- Single user
- Stateful connections

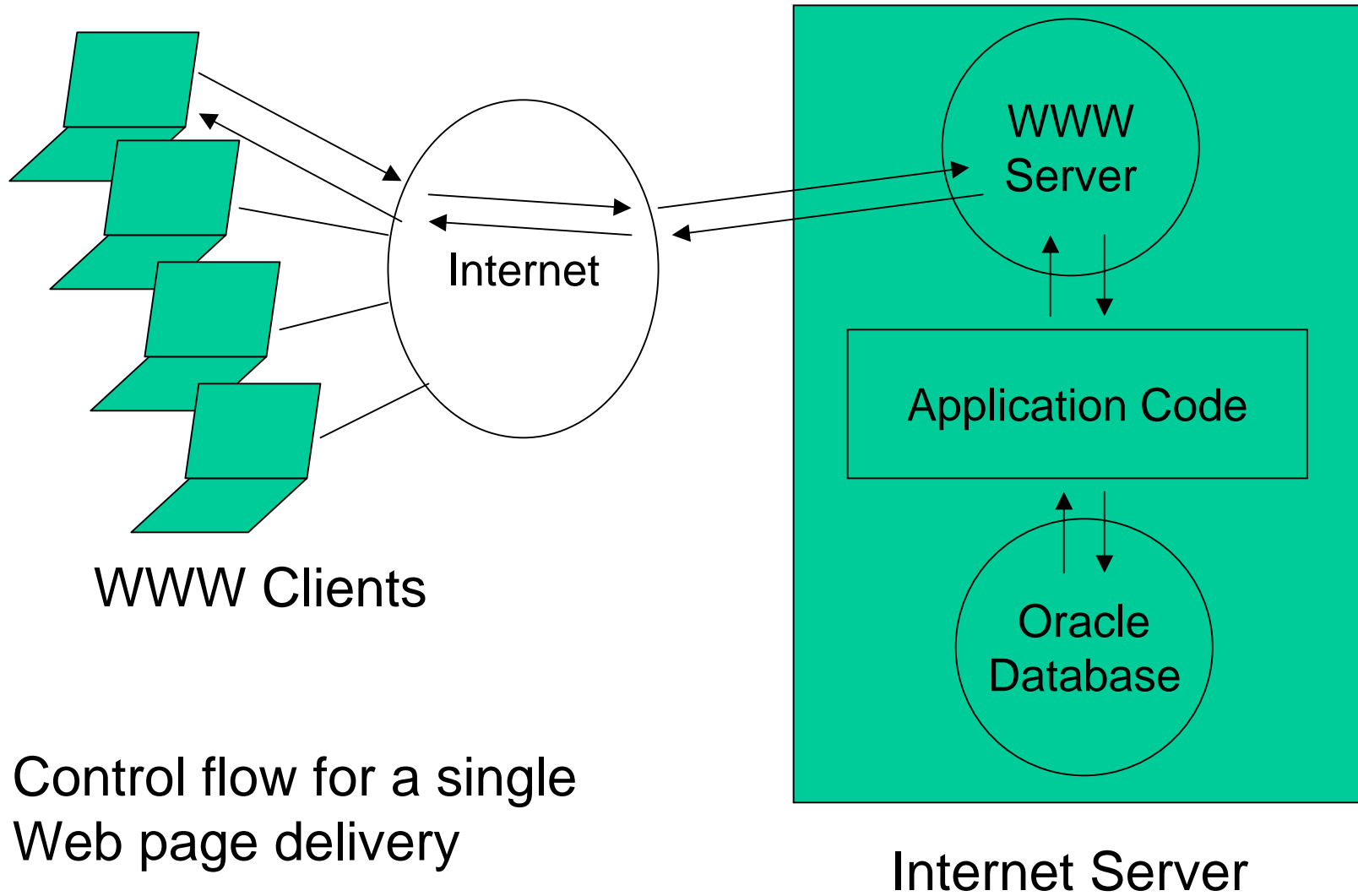
Internet Programming

- Page-Click-Page
- Program runs on remote server
- Slow Internet connection
- 10.000 concurrent users
- Stateless browser

Summary

- OO programming well suited to handle events: events -> methods
- No events, no OO...

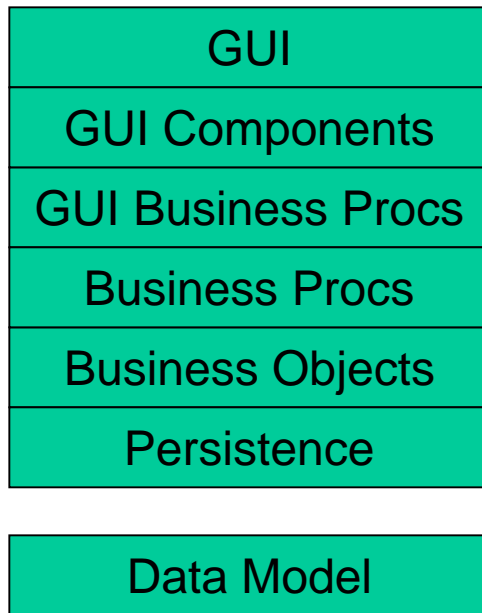
ACS in its Context



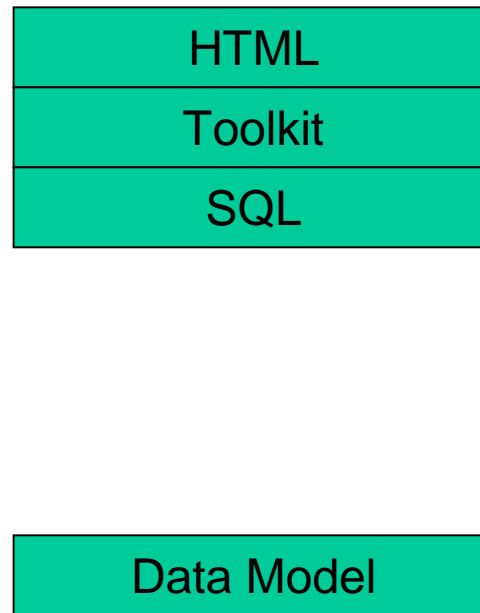
Control flow for a single
Web page delivery

Java Against ACS Layer Cake

OO (Java)



ACS



Other Web Architectures

- Apache + Java + Jserv (Jakarta)
- Apache + Perl
- Apache + PHP
- Application Servers
 - Oracle OAS
 - IBM Websphere
 - BEA Weblogic
 - Netscape NAS
 - ...

But none of the systems contains ready to use modules.

Why ACS?

- Comes with a large number of ready-to-use modules
- Highly efficient programming model:
 - “Page-Click-Page” interaction model
 - Database oriented (not object oriented)
 - SQL statements close to HTML code
- It works. Even with a large number of users.
- Comes with lots of documentation
- Comes with philosophy that makes producing results easy.

Reliable Web Services

- Need a relational database as backend. Current OO database do not provide:
 - High performance (100 concurrent users)
 - Transaction isolation
 - High availability
 - Online backup
 - Online replication
 - Administration tools
 - Remote access

3. Case Study

Case Study

- Driving School Portal
- Target Group
- Why Join the Portal?
- Additional Contents
- Which ACS Modules?
- Marketing
- Make Money
- Why didn't I do it myself?

Driving School Portal

Idea

- Make a portal for all people who have to learn driving a car or motorcycle

Target Group

- 10.000 driving students in Barcelona every year
- Age between 18 and 22
- Both sexes equally
- From all social layers
- Students, young professionals, ...

Why Join the Portal?

- Compare driving school prices
- Compare failure rates
- Compare driving school quality
- Buy literature
- Talk about exercises
- Chat with driving school experts
- . . .

Additional Contents

- Buy a used/new car
- Meet a boyfriend/girlfriend
- Look for a job
- Share apartments
- Party calendar
- Buy accessories
- . . .

Which ACS Modules?

ACS provides ready to use modules

- Static Contents
- Opinion Polls
- Bulletin Boards
- Chat
- Classified Ads

Marketing

- Tell your friends and tell them to tell others
- Publish articles in local press
- Setup posters in the University
- Let best rated driving school make publicity with the rating
- Get an entry in the Yellow Pages
- See <http://www.gorillamarketing.com/>
- . . .

Make Money

- Sell ad space to local driving schools
- Sell ad space to general marketing agencies
- Ask for commission to sell used cars
- Partner with book store to sell literature
- Make homepages for driving schools
- . . .

Why didn't I do it Myself?

- Literature is free in Barcelona.
- In Belgium (that's where the idea comes from), the driving school book costs 40.000pts
- I have no friends who are in driving school

4. How To Build Your Portal?

How to Build Your Portal?

- Become an ACS developer
- Get an ACS Server
- Build a Simple ACS Portal
- Build Custom Modules
- Form Your Community
- Make Money

Become an ACS Developer

What to do?

1. Learn some Linux
2. Learn TCL
3. Learn SQL

How to do?

- Install Linux at home
- ACS problem set 1
- ACS problem sets 1 & 2

Get together with some friends who have done it already

Get an ACS Server

What to do?

1. Get a Linux server in the Internet
2. Get an Oracle license
3. Install Oracle
4. Install ArsDigita

How to do?

- ADSL, University Cluster Competitiveness?
- Evaluation license is free
- Ask friends who did
- Follow ACS instructions

Build a Simple ACS Portal

1. Get an idea of what you want to build
2. Define a web design
3. Configure some existing modules
4. Make some small changes to the modules
5. See why nobody is using your portal
6. GOTO 1

Build Custom Modules

1. Get an idea of what you want to build
2. Define a web design
3. Make a “Wemo” (=Workflow Demo) for new modules
 - Present the Wemo to friends & family.
 - The Wemo will save a lot of time during development.
4. Configure some exiting modules
5. Make an interaction model
6. Make a data model.
7. Write the TCL pages
8. Test the system together with some friends
9. See why nobody is using your portal
10. GOTO 1

Form Your Community

What to do?

1. Setup an initial community
2. Get killer content into your portal
3. Attract more users
4. Assure high quality

How to do?

- Tell your friends to participate
- Convince your friends to attribute interesting stories etc.
- Assure high quality/usability
- Remove old/bad contents

Make Money

- Making money with a portal today is nearly impossible.
- You can try to sell your portals to people who still believe they can make money...

5. Conclusion

- ACS is the tool of choice to create real portals
- ACS is easy to learn
- The experiences made with ACS are useful even for all Web environment

6. References

Related Literature

- Ars Digita: <http://www.arsdigita.com/>
- TCCG: <http://www.competitiveness.com/>
- ACS Documentation: <http://www.arsdigita.com/doc/>
- The Online Bible:
<http://www.arsdigita.com/books/panda/>