Competitiveness.com
Building Competitive Business Clusters

# Building ArsDigita Portals #3

Frank Bergmann <fbergmann@competitiveness.com>

Barcelona, March 1st, 2001

## Content

- Recap: ACS Architecture
  - Reliable Web Services
  - AOLServer
  - AOLServer against Apache
  - ACS Application Architecture
- How To Build Your Portal
  - Define the Project
  - Setup an ACS Server
  - Build a First ACS Portal
  - What Went Wrong?
  - Form Your Community
  - Build Custom Modules
  - Make Money

## 6. Structuring Techniques

Questions:

- How are large applications structured?
- How to avoid duplicated code?

Answer:

- Methods & Procedures
- Objects, Classes & Modules
- Class Models & Data Models
- Patterns

## Structuring Techniques

### Java

- Object/Classes/Methods
- Patterns: help to arrange larger functionalities
- Java Beans
- Modules consist of 1 or more classes
- The data model is hidden from the implementation by the persistence layer

### ArsDigita

- Pages serve as “methods”
- Objects reside in the database
- Common functionality is “factored out” to TCL procedures
- The data model is in the center of the application

## 7. Development Methodology

- How to organize large projects?
- How to avoid that very “clever” guys make the same mistakes over and over again?

## Java Against ArsDigita Development

### Java

- Write **Use Cases** using UML
- Identify **Components** to be implemented using **Java Beans**
- Make your system design using **Patterns**
- Create a **Class Model** using Class Diagrams
- Create a **Data Model** from the Class Model
- Define important **Methods** for each Class using Collaboration Diagrams

### ArsDigita

- Write **Use Cases** with or without UML
- Identify **Components** to be implemented using **Procedures**
- Create a **Data Model**

## Application Design

- GUI and User Interaction
- Data Model / Object Model
- Business Logic
- Architecture

## GUI Development

### ACS

- Focus on page flow
- Implemented as a “Wemo” using static HTML

### Java

- Focus on Forms and interactive elements in those forms
- Implemented using RAD (Visual Basic, Visual Age for Java, …)

## Object/Data Model

### ACS

- Object reside in the database
- The Data Model corresponds to the Object Model

### Java

- Object model using UML Class Diagrams
- Data model using ERP modeler
- Integration using a persistence module/manager

## Business Logic

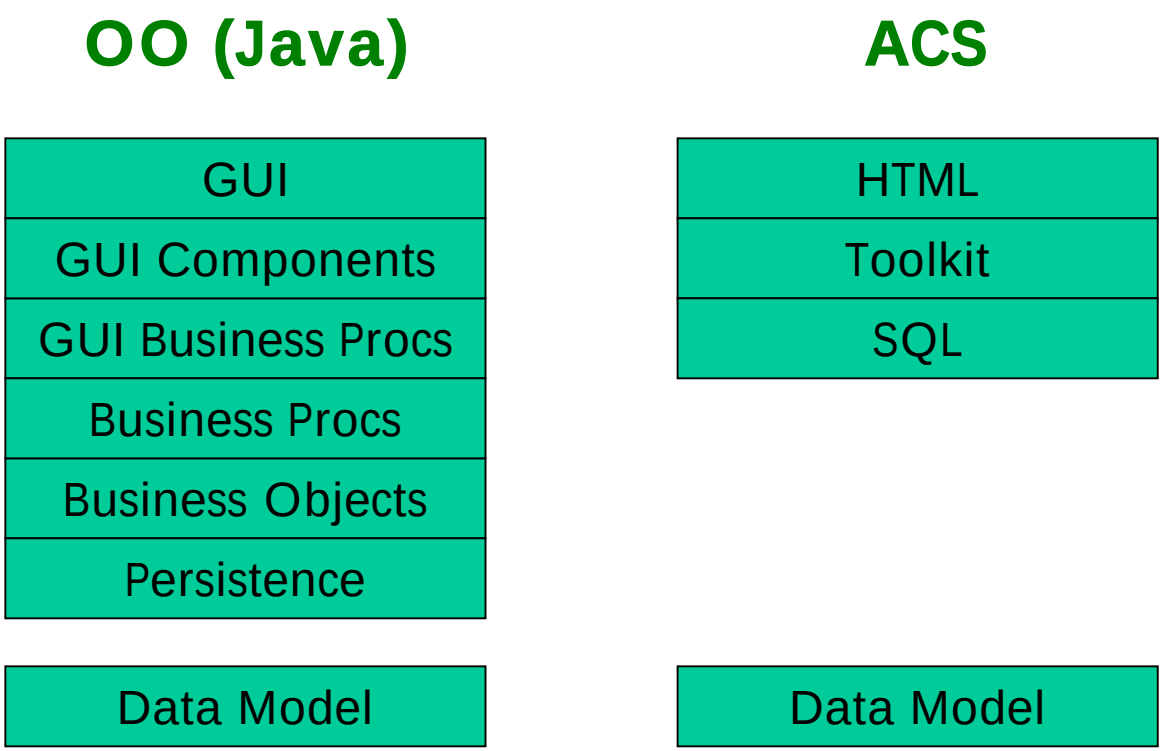
### ACS

- Business Logic integrated into TCL Pages

### Java

- Separation of Business logic into a separate layer

# Java Against ACS Layer Cake



# How to Build Your Portal?

- Define the Project
- Setup an ACS Server
- Build a First ACS Portal
- What Went Wrong?
- Form Your Community
- Build Custom Modules
- Make Money

# Driving School Portal

## Case Study

- Idea
- Target Group
- Why Join the Portal?
- Additional Contents
- Which ACS Modules?
- Marketing
- Make Money

## What to do?

1. Learn some Linux
2. Learn TCL
3. Learn SQL
4. Get a Linux server
5. Install ArsDigita
6. Install Oracle

## How to do?

- Install Linux at home
- ACS problem set 1
- ACS problem sets 1 & 2
- PC with 128MByte RAM
- Read online doku
- Read online doku

Get together with some friends who have done it already

## Build a First 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
   or continue with next slide

## What Went Wrong?

- “Empty bar effect”: Nobody likes to enter an empty bar”:
  - Create artificial “noise”
  - Ask your friends to participate
  - Actively form your community

- “This is not my bar effect”: Nobody likes to enter a bar which is not his/her background
  - Check that application modules 100% fit your needs
  - Analyze in detail user behavior (ask your girlfriend/boyfriend)
  - Build custom modules

| What to do? | How to do? |
| --- | --- |
| 1. Setup an initial community | ▪ Tell your friends to participate |
| 2. Make people stay in your portal | ▪ Get killer content<br>▪ Design apps for people to stay<br>▪ Make “Strategic Partnerships”<br>▪ Import contents from other sites |
| 3. Attract/maintain users | ▪ Assure high quality/usability<br>▪ Remove old/bad contents |

## 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

## 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…

## 9. 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/