Sports Management Information Systems

Camilo Rostoker
November 22, 2002
Introduction

- We are in the information age

- The availability of technology has brought forth a new problem domain – how do we manage the nearly limitless amount of information?

- Solution: In the broadest term, information technology. More specifically, we look at the use of Information Systems
Information Systems

- **What is an information system?**

  A system that is designed to capture, transmit, store, retrieve, manipulate, and/or display information used in one or more business processes.

- **Information Systems are central to Information Technology**
Now, IS covers a broad spectrum of application domains.

The scope and complexity of IS’s today vary from small utility systems, to complex mission-critical enterprise level systems.
Examples of Information Systems:

- Geographic Information Systems
- Library Information Systems
- Learning Information Systems
- Enterprise Information Systems
- Management Information Systems
The most common use of Information System is for business management purposes – Management Information Systems (MIS)

With the push for I.T. to be the backbone for businesses, MIS’s have been extensively researched as a highly effective tool for management, audit & control of information within a business.
The technology to use is highly dependant upon the architecture of the Information System.

Global cooperation requires a completely neutral and platform independent architecture.

The main technology leading this push is XML – an extensible markup language.
Basic Example:

```xml
<TeamInfo>
  <TeamID> 255 </TeamID>
  <TeamContact>
    <ContactName> Camilo Rostoker </ContactName>
    <ContactEmail> camilo@scottsdale.ca </ContactEmail>
  </TeamContact>
  <TeamWins> 9 </TeamWins>
  <TeamLosses> 3 </TeamLosses>
  <TeamRoster>
    <Player>
      <PlayerID> 2845 </PlayerID>
      <PlayerName> Jack Kanaska </PlayerName>
    </Player>
    <Player>
      <PlayerID> 3283 </PlayerID>
      <PlayerName> Billy Bob </PlayerName>
    </Player>
  </TeamRoster>
</TeamInfo>
```
In particular, this presentation will focus on the analysis and design of an Information System for sports organizations.

A sports organization is like any other small business - it contains:
- business processes, information management and control
- provide a product to its customers
Information Systems for Sports Organizations

Therefore an IS for a sports organization can potentially:

- improve information management, leading to more efficient business processes
- reduced administrative workload
- improved access to information for both management and members
- Provide highest Quality of Service (QoS) to the members of the league
Objectives & Goals

- Model organization structure and key administrative activities
- Be useful not only to management, but also to members of the organization
- Improve administrative efficiency
- Reduce administrative workload
- Improve communication between management and members
- Contribute to success of organization’s continued growth and development
- Provide useful tools & services to both management and members of the league
- Increase member satisfaction
- Improve Quality of Service (QoS) to members of the organization
Available Solutions

- **LeagueWorks Software**
  http://www.leagueworks.com

- **Analyzer Software**
  http://www.analyzersoftware.com

- **EventScores**
  http://www.EventScores.com

- **All American SportsWare**
  http://www.allamericansportsware.com/
WebCoach – A Sports Management Information System

- An all-in-one solution to sports league management

- A prototype IS demonstrating parts of solutions to the specified requirements

- Been in continuous development for over 2 years
History of WebCoach

- Original concept began when serving on RUFDC executive board

- Now, it’s RUFDC’s primary tool, and along with league website, are a major reason for the continued growth and development of the league

- A perfect environment for the analysis and design of such a system.
Analysis & Design

- Analysis and Design from a Software Engineering perspective

- Software Engineering: “used to narrow the gap between the given problem and their solutions.”

- Little documented work on SE approaches to SMIS
Requirements Analysis

- 3 main steps in requirements analysis:
  
  1. Requirements Elicitation – understanding the problem at hand
  2. Requirements Specification *
  3. Requirements Validation and Verification
Requirements Elicitation

Understanding the problem at hand:

- Issues in providing efficient administration due to lack of process, resources, people, money, etc.

- Issues in communication between administration and member base

- Result: inability to provide quality service to members of the league, waste of available resources, little organizational growth
Requirements Specifications

- Obtaining Requirements Specifications:
  Decompose system into subsystems
Subsystems Specifications

- Different requirements for:
  - Public & Guests
  - Team Administrators
  - League Administrators
  - System Administrators

- Generic Requirements
Process Models

- In developing a system model, there are 3 steps required:

  1. Analyze current business processes and obtain the physical model of it
  2. Obtain logical model of current system from physical model
  3. Create improved logical model of target system
Physical Model

Sample Process – Submitting & Recording Game Results

- Team Captain
- League Admin
- Excel Spreadsheet
- Website files (HTML)

Flow:
1. Team Captain sends result to League Admin via email or phone.
2. League Admin records result and re-calculates standings.
3. League Admin updates standings on the website.
4. Save result in Excel Spreadsheet.
5. Website files (HTML) are saved and updated.
Logical Model # 1

Team Captain

Record Result

Process Game Result

Recalculate Standings

Excel Spreadsheet

Website files (HTML)

Update Team stats

New game record

Update Standings on Website

New standings
Development Process Model

- Combination of incremental and prototype

  - **Incremental**: have released 3 working version of the system, each one containing new and improved functionality

  - **Prototype**: system is meant to demonstrate the usefulness and functionality
## Design Issues

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Other Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Languages &amp; Environment</td>
<td>Flexibility</td>
</tr>
<tr>
<td>Data Store and Retrieval</td>
<td>Scalability</td>
</tr>
<tr>
<td>User Interface</td>
<td>Portability</td>
</tr>
<tr>
<td>Concurrency</td>
<td>Testability</td>
</tr>
<tr>
<td>Exception/Error Handling</td>
<td>Performance</td>
</tr>
</tbody>
</table>
Language & Environment

- Main development language: Perl

- Advantages:
  - Convenient for web-based systems and prototypes
  - Easy to learn and use for simple applications
  - Scripting language → does not need to be compiled
  - Portable → can be ported to other systems, including Windows

- Disadvantages:
  - Slow, must be processed by interpreter first
  - Lack of support for object-oriented analysis & design
  - Hard to debug, no visual IDE’s

- Other development languages: JavaScript, D/HTML, CSS

- Environment: currently on a Linux Red Hat system, running Apache web server
Data Storage & Retrieval

- Main data is stored on a Database
- Also use plain-text files for some less frequently accessed data

Database:
- MySQL
- DBI, a Perl API for MySQL

MySQL DBMS is a free product, is reliable, so a good choice for prototype implementations of Information Systems
User Interface

- Displayed using dynamically generated HTML
- Graphically and visually enhanced using CSS, icons, color schemes, etc.
- Layout resembles typical webpage format for easy navigation (ie: menu on left)
- Language (ie: text, labels, buttons) is clear and informative, to help even the most technically challenged people
- Appearance can be partially customized by the league admin
  - Good for matching existing website or league theme
  - However, this causes some problems. Issues on level of customization that should be provided
- Enhanced for Internet Explorer, however still works with Netscape. Other browsers may not be compatible.
Concurrency

- Very important due to the amount of potential concurrent accesses
- Perl interpreter – spawns new process to handle each new request
- MySQL DBMS - provides record locking support for concurrency control
- Files – all access to files use a file locking mechanism to signal shared or exclusive locks on files
- Concurrency is hard to test
- Not so crucial now, but will become more critical as number of users increase and additional functionality is added, such as online payments, etc.
Exception/Error Handling

- Some checking done on client-side (ie: input validation) using JavaScript

- How to deal with:
  - System Errors
  - Other non-system errors

- Provide facilities for error and bug reporting
Since system is designed to be used by various sports organizations, it is important for the system to be flexible enough to suit particular league needs.

For example, the league admin can customize their system using the “Preferences” page.

- Can customize such settings as:
  - appearance,
  - Main page content options
  - Game modes
  - Message footers

Also, since the system is essentially displayed like a webpage, it can be integrated into existing sites, or run stand-alone in its own window.
Scalability

- Any system that expects an increase in use should be scalable.
- Should be able to handle a few users, or thousands of users.
- MySQL is designed for scalability and can easily handle thousands of records in a single table.
- Design of the database itself is also important.
Portability

- Portability is a key design consideration, especially when prototyping.
- Technologies used for WebCoach prototype allow for easy portability across platforms.
- Presentation Logic (UI) is on client side, so don’t need to worry about that.
- DB access logic is standard SQL, so should port to various other DBMS’s.
Testability

- Some modules of the systems can be tested manually, with immediate results

- Allows for convenient testing of functionality by created “mock” leagues and simulating inputs and possible user actions

- Currently have a few leagues using the system

- However some test can only be incurred with continuous use and extended time
Performance

- Doesn’t need to be lightning quick, but shouldn’t run unnecessarily slow either, or users will become annoyed with the system.

- Current system is fairly slow, due to:
  - Server hardware
  - Network connection
  - Perl interpreter
  - Efficiency of algorithms
Security

- Need to secure system against malicious hackers

- Keep the organizations (and members) data confidential (as outline in WebCoach Privacy Policy)

- Currently, data stored on WebCoach is not highly sensitive (ie: no credit card information or other important person info)

- However, future additions may require higher security, or just the ability to guarantee confidentiality will require higher security
For example, a major security feature of WebCoach is Persistent Session Management.

- Uses concept of Session ID (SID), a unique number that remains with a user for the entire period of his/her session on the system.

- Internally, SID is linked to users information, such as username, password, account information, session information, etc.

- Password is NEVER displayed on the screen at any point.

- Time tracking within the SID allows for “Idle Time” expiry of sessions.

- Can determine when and where user last logged on from.
**Session Management**

**Login Request**
- Login
- Authorize
- **Authorization Successful?**
  - NO
  - YES: Generate Session ID
- Update Session ID
- Default Home Page

**Any other Request** (User is already logged in)
- Any Request
  - Validate Session ID
  - **Error Message (Login Again)**
    - NO
    - YES: Update Session ID
  - Re-direct to Requested Module/Service
Evaluation

- Need some criteria to from which to evaluate.

- Methods of Evaluation:
  - Based of previously set goals and objectives of target system
  - Based on feedback
    - Random comments and feedback
    - Results from recent survey
Issues With Current Design

- Maintenance
  - Dev site uses the same database as public site...should use a mirrored copy of the DB instead
  - Version and module upgrading not automatic
    - Have to manually copy over code and make minor modifications to script to include new modules
    - Should be an automated process to discover new modules
  - No automated database backups
Issues With Current Design

- Security
  - Perl, easy to hack unless properly coded
    - code is probably full of vulnerabilities
    - Would need an experienced security consultant/analyst to review code and overall design

- MySQL privileges
  - Current design uses only one account to access the DB
  - Permission and access controlled by program through authorization
Looking Forward

Ideal situation for a production-ready WebCoach would include:

- Modifying design to fix a few of the flaws
- Implementing on an Enterprise Application Server using Java technology
- Would allow the integration of cutting-edge Java technologies such as:
  - XML, including SOAP
  - Java Servlets & Java Server Pages (JSP)
  - Enterprise JavaBeans (EJB)
  - JDBC
- Java would be faster, very secure, more reliable, potential to include more complex functionality
Conclusion

- WebCoach was conceived because there was a need for it – a problem without a solution.

- WebCoach has several issues with it, but also contains many positive aspects.

- WebCoach has been a learning experience for me → experience with analysis, design and implementation of extensive information system.


3. *Requirements Engineering & Expectations Management, and the Two Cultures*. Barry Boehm, Marwan Abi-Antoun, Dan Port, Julie Kwan, and Anne Lynch, University of Southern California