ACMS

Version 1.0.1

A System for Academic Course Management and Scheduling via the World Wide Web
Problem Statement

- No centralized repository for University calendaring and scheduling information.

- No easy means for faculty to create, maintain and publish information to a centralized, electronic information repository, for example, academic course information via the World Wide Web.

- No easy and centralized resource for individuals across campus wishing to view all of their schedule or calendaring information
Data Flow Diagram

ACMS Client

Authentication Interface

Acceptance or Rejection

Identification

Kerberos

Main Interface

Client Data

ACMS Server

Query

Database

Query Result

Results
Authentication Interface

- **Kerberos Principal**
- **Kerberos Password**
- **Kerberos Relay Server**
  - Authentication Interface

- **Authentication Server**
- **Ticket-Granting Server**
  - **Kerberos Key-Distribution Service**

- **Authenticated?**
  - Yes: Determine Group Associations
    - Select Front-End Interface and Proceed with Client Applications
  - No: Abort
ACMS Server Interface

Event Information Management

Front-End Application

User Commands
  Filter Query
  Search Query
  Event Add
  Event Modify/Delete

RDBMS

ACMS Server

Event Returns
  Embedded Hypertext Events
  Email (SMTP) Events
  ACMS Event Format
  ACMS Status/Error
ACMS Server Interface

Course Website Management

Front-End Application

User Commands
- Show Events
- Event Add
- Event Modify/Delete
- Transaction Commit
- Transaction Undo

RDBMS

ACMS Server

Server Actions
- Transaction Logging
- Data Recovery
- Website File Management
- ACMS Status/Error

Transaction Commit

Transaction Undo
Hardware and Operating System Specs

ACMS will consist of:

1. A Solaris 2.5 SPARC machine running NCSA httpd 1.5a. ACMS client software will be distributed via httpd to client machines.

2. A server of to-be-determined specifications running Sybase 11.0.1 as the core ACMS data repository back-end.

ACMS client software will require a Java capable environment. ACMS shall be written under the assumption that all client machines will be Java capable.

Java front-end applications will be written using Version 1.1 of the Java Development Kit.
General Implementation Specs

1. Languages: Java, C++ and Perl.

2. javadoc utility conformity; APIs and documentation will be written for all interfaces.

3. vCalendar specification conformity.

4. All time and date references will adhere to ISO 8601 standards.

5. ACMS will be written to support any character set registered with the Internet Assigned Numbers Authority (IANA), defaulting to the ASCII character set. ACMS will consequently support any language as specified in the vCalendar specification.
**Potential Risk/Problem Areas**

1. Efficient user and group management, user population size.

2. Applet code signing and related security issues.

3. Database reliability and recoverability.

4. Network and server reliability