

**MICHIGAN STATE UNIVERSITY**

## 6. & 7. Teams: Technical Specifications and Schedule

CSE 498, Collaborative Design



Wayne Dyksen  
Brian Loomis  
Department of Computer Science and Engineering  
Michigan State University  
Spring 2006

**MICHIGAN STATE UNIVERSITY**

## 6. & 7. Teams: Technical Specification / Schedule Auto-Owners Insurance

Team 1: Auto-Owners Insurance  
CSE 498, Collaborative Design



Jason Tuck  
Phil Danne  
Ferenc Pierre Szabo  
Joe Blam  
Department of Computer Science and Engineering  
Michigan State University  
Spring 2006

**S Project Overview**

Team 1: Auto-Owners Insurance

- Annuity Overview
  - Someone has an insurance claim or comes across some external large sum of money
  - Auto-Owners offers long-term investment schemes
- Existing Program Overview
  - 1980 Basic Program – plan details computed
  - Some calculations done by hand
  - Lotus Program – data re-entry for presentation
  - Hard copy sent to be re-entered into mainframe
- New System Specs
  - Enter Information of Claimant
  - Generate annuity plan presentations
  - Get response/selection from claimant
  - Retrieve proposed plans from data base
  - Send info to mainframe

4-3

**S Architecture Components**

Team 1: Auto-Owners Insurance

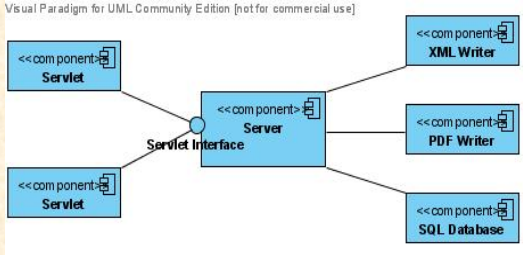
- Hardware Platforms
  - Java Application run on unspecified web server
  - Generates web pages for arbitrary client computer
- Software Platforms / Technologies
  - Java Servlets run on websphere application server
  - html forms output to client
  - JDBC interface used to run SQL on DB2
  - XML output generated to be sent to the server
  - Formatted presentation of data generated (html) & converted to a pdf file

4-4

**S Architecture Illustrated**

Team 1: Auto-Owners Insurance

Visual Paradigm for UML Community Edition [not for commercial use]



```

    graph LR
        S1[Servlet] --- SI[Servlet Interface]
        S2[Servlet] --- SI
        SI --- Server[Server]
        Server --- XML[XML Writer]
        Server --- PDF[PDF Writer]
        Server --- SQL[SQL Database]
    
```

4-5

**S Architecture Risks**

Team 1: Auto-Owners Insurance

- Understanding Legacy System
- Understanding Legacy Code Base
- Java, WebSphere, Servlets

4-6

**S Project Schedule**

Team 1: Auto-Owners Insurance

1. Determine Platform
  - a) Goal: Determine server & languages used
  - b) Date: 1/25/06
2. Understand Legacy Program
  - a) Goal: BASIC code, Lotus 123
  - b) Date: 1/25/06
3. Design New System
  - a) Goal: Classes, Interactions
  - b) Date: 2/1/06
4. Division of Development Tasks
  - a) Goal: Equal work for all members
  - b) Date: 2/1/06

4-7

**S Project Schedule**

Team 1: Auto-Owners Insurance

5. Coding & Integration
  - a) Goal: Writing actual program
  - b) Date: 3/3/06
6. Unit Testing & Debugging
  - a) Goal: Get everything working!
  - b) Date: 3/24/06
7. Additional Features?
  - a) Goal: Make the program better!
  - b) Date: 4/1/06
8. Debugging/Integrating Additional Features
  - a) Goal: Get everything working again...
  - b) Date: 4/15/06

4-8

**S Project Schedule**

Team 1: Auto-Owners Insurance

9. Presentation
  - a) Goal: Present final product to AOI
  - b) Date: 4/17/06
10. Stay in Contact
  - a) Goal: Ensure no last-minute changes needed
  - b) Date: 4/24/06

4-5

MICHIGAN STATE UNIVERSITY

6. & 7. Teams:  
Technical Specification / Schedule  
GM Strategy Generator

Team 2: ChannelVantage  
CSE 498, Collaborative Design

«Jake Lunetta»  
«Matthew Kopchick»  
«Laron Myers»  
«Nam Nguyen»

Department of Computer Science and Engineering  
Michigan State University  
Spring 2006



**S Project Overview**

Team 2: ChannelVantage

- Port the current application to a web-based one (using ASP.NET) that can automate the previous process of manually updating the GM mainframe with pre-define legal strategy text.
- Completely redesign the UI so that it is clean, and user-friendly while adhering to the GM DealerWorld standard
- Add User/Groups that have specific abilities of their own, allowing a large number of users
- Add report viewing functionality that is compatible with the new Web application

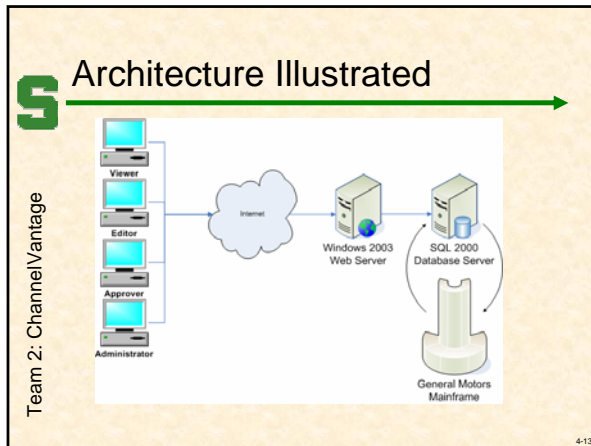
4-11

**S Architecture Components**

Team 2: ChannelVantage

- Hardware Platforms
  - Any x86 architecture
- Software Platforms / Technologies
  - Windows 2003/XP
  - SQL Server 2000
  - ASP.NET, C#, SQL, JavaScript
  - IIS
  - HTML, XML, CSS
  - ActivePDF or iTextSharp

4-12



- ### Architecture Risks
- Making a UI that is completely redesigned yet has a low learning curve.
  - Make sure that all aspects of the Web application comply with GM DealerWorld Standards
  - Graceful error checking and reporting
  - How to store temp data when users are changing between UI fields.
  - Figuring out how to generate PDF files

- ### Project Schedule
1. Setup Workstation/Server
    - a) Goal: Have appropriate software installed
    - b) Date: 1/20/2006
  2. Meet with client
    - a) Goal: Get a set basis of requirements/Information
    - b) Date: 1/17/2006
  3. Dissect the database
    - a) Goal: Understand the Client's table layout
    - b) Date: 1/26/2006
  4. Create a new database schema
    - a) Goal: To have a functional Relational Database
    - b) Date: 2/2/2006

- ### Project Schedule
5. UI mockup
    - a) Goal: Draw up UI in DealerWorld stylesheet
    - b) Date: 2/2/2006
  6. Figure out PDF software
    - a) Goal: Choose a PDF package for reporting
    - b) Date: 2/2/2006
  7. Final Accepted UI design
    - a) Goal: Client accepted design
    - b) Date: 2/6/2006
  8. Working Prototype (MDA UI)
    - a) Goal: Have a prototype for class presentation
    - b) Date: 2/20/2006

- ### Project Schedule
9. Working SDA UI
    - a) Goal: Have SDA interface working
    - b) Date: 2/27/2006
  10. Working Reports
    - a) Goal: Printing and approving reports functional
    - b) Date: 3/4/3006
  11. Working Administration Tools
    - a) Goal: Admin functionality useable
    - b) Date: 3/20/2006
  12. Final Presentation
    - a) Goal: Working product/documentation
    - b) Date: 4/24/2006



**6. & 7. Teams:**  
**Technical Specification / Schedule**

Team 4: Identity Alliance  
CSE 498, Collaborative Design

Scott Donahue  
Dani Duchesne  
Tom Hanning  
King Leung

Department of Computer Science and Engineering  
Michigan State University  
Spring 2006



## S Architecture Components

- Hardware Platforms
  - Hardware Independent
- Software Platforms / Technologies
  - Firefox
  - Internet Explorer
  - MySQL
  - ID4Sure Framework ( ActiveX or Firefox Extension)
  - AJAX (JavaScript and XML)

Team 4: Identity Alliance

4-19

## S Project Overview

- Technical Specs Complete
- Updated Framework API Document
- Framework Delivery by Feb 3rd
- Generating Web pages and setting up AJAX connections.

Team 4: Identity Alliance

4-20

## S Architecture Illustrated – part III

### Document Management using ID4Sure Framework

The diagram illustrates the document management process. On the left, a 'Card Holder' (represented by a person icon) and an 'SC Reader' (represented by a green box) are connected to a 'Doc Manager (ours)' (blue vertical bar). This connects to the 'ID4 Framework' (blue vertical bar), which then connects to a 'Web Browser' (blue vertical bar). The Web Browser connects to a 'Server' (orange vertical bar) which contains a 'Document Repository' (pink box) and a 'Possible Queueing System' (pink box).

Team 4: Identity Alliance

4-21

## S Architecture Risks

- Incompatibility with web browsers (Firefox and IE only)
- Limited data space on smart cards
- Retrieving and sending data to smart card
- Utilizing checksums in project 3
- Framework bugs and limitations

Team 4 – Identity Alliance

4-22

## S Project Schedule

1. Project Technical Specification Submission
  - a) Goal: Submit Technical Specification Document
  - b) Date: 1/29 - completed
2. Tech. Spec. Client Submittal
  - a) Goal: Submit Technical Specification to Client
  - b) Date: 1/31 – completed
3. Prototyping project 1
  - a) Goal: Finish prototyping portal
  - b) Date: 2/7
4. Prototyping project 1 milestone 2
  - a) Goal: Show client prototype and obtain feedback
  - b) Date: 2/9

Team 4: Identity Alliance

4-23

## S Project Schedule

5. Prototype Project 2
  - a) Goal: Prototype Project 2 complete
  - b) Date: 2/16
6. Prototype Project 2 Check-in
  - a) Goal: Show the client our progress and get feedback
  - b) Date : 2/20
7. Prototype Project 3
  - a) Goal: finish prototype of project 3
  - b) Date: 3/18
8. Prototype Project 3 Check-in
  - a) Goal: Get Feedback – add features?
  - b) Date: 3/21

Team 4: Identity Alliance

4-24

## Project Schedule

Team 4: Identity Alliance

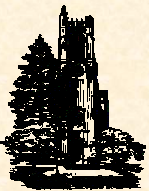
5. Final Testing Phase
  - a) Goal: Finish all testing on Project 1
  - b) Date: 4/1
6. Final Testing Phase Project2
  - a) Goal: finish testing on project 2
  - b) Date: 4/7
7. Final Testing Phase Project 3
  - a) Goal: Finish testing on Project 3
  - b) Date 4/14
8. Final Check-in and approval of entire system
  - a) Date: 4/21 - lots of time at the end for error and to add optional features

4-25

**MICHIGAN STATE UNIVERSITY**

## 6. & 7. Technical Specification / Schedule Image Space Telemetry Data Viewer

Team 5: Image Space  
CSE 498, Collaborative Design



Manju  
Phil  
Matt  
Nick  
Department of Computer Science and Engineering  
Michigan State University  
Spring 2006

## Project Overview

Team 5: Image Space

- Telemetry Data is complex, dumb it down for me.
- It's a game, needs to be simple.
- Data source needs to be flexible, post-time and real time.
- Charts and graphs to simplify complex barrage of data.
- Allow graphs/charts to be viewed simultaneously to view trends between data.

4-27

## Architecture Components

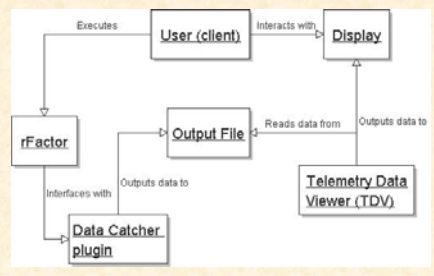
Team 5: Image Space

- Hardware Platforms
  - X86 based PC.
  - Reasonable video card, Geforce 3/Radeon 8500 or higher required for game play.
- Software Platforms / Technologies
  - DirectX.
  - Windows 2000/XP.
  - TCP/IP for real time data viewing.

4-28

## Architecture Illustrated

Team 5: Image Space



4-29

## Architecture Risks

Team 5: Image Space

- Biggest risk is lack of DirectX experience
- Performance, data file contains snapshots of data from every ~.025 seconds. Over a race this is a lot of data to compute with. An average lap time of ~3 minutes this is 7200 snapshots of data for an average lap. To make it worse, during intense parts of the game it outputs every .011seconds, creating more snapshots.
- Switching data source from post-time to real-time.

4-30

**S** Project Schedule →

Team 5: Image Space

4-31

**S** Project Schedule →

Team 5: Image Space

1. Test
  - a) Goal: Pre-spring break test.
  - b) Date: 3/2/06
2. Relax
  - a) Goal: Stop having nightmares of F1 cars screaming by.
  - b) Date: 3/10/06
3. Resizeable/Scalable windows
  - a) Goal: Allow different orientations of view ports.
  - b) Date: 3/16/06
4. Additional Data Displays
  - a) Goal: Display more details, allow drill downs?
  - b) Date: 3/16/06

4-32

**S** Project Schedule →

Team 5: Image Space

1. Real-time
  - a) Goal: Switch data source to real-time source.
  - b) Date: 3/30/06
2. Test
  - a) Goal: Stress test/ data verification.
  - b) Date: 4/30/06
3. Pretty++
  - a) Goal: Play with DirectX prettying functions.
  - b) Date: 4/30/06

4-33

MICHIGAN STATE UNIVERSITY

6. & 7. Teams:  
 Technical Specification / Schedule  
 P2P Applications and Controls Project

Team 6: Microsoft P2P  
 CSE 498, Collaborative Design



Carly Szekely  
 Tom Vollman  
 Dan Lash  
 Mike Fazio

Department of Computer Science and Engineering  
 Michigan State University  
 Spring 2006

**S** Project Overview →

Team 6: Microsoft P2P

- Build a class library to make P2P chat room development easier
- Encapsulate low level P2P calls into API
- Geared towards building tools not a single application
- Building block between low level Windows calls and high level abstraction

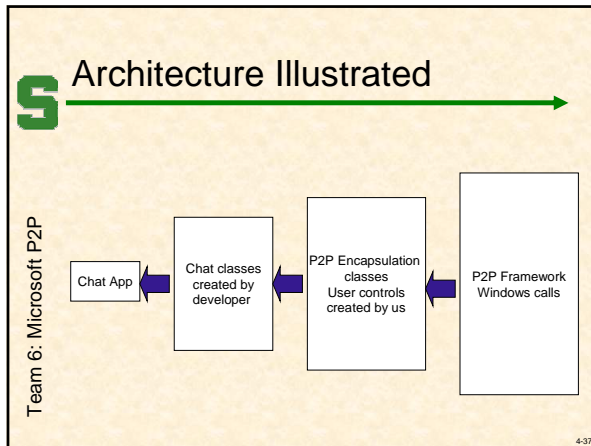
4-35

**S** Architecture Components →

Team 6: Microsoft P2P

- Hardware Platform
  - x86
- Software Platforms / Technologies
  - Windows XP SP2
  - Windows Server 2003
  - Vista
  - Visual Studio 2005
  - .Net 2.0
  - C# 2.0

4-36



- ### Architecture Risks
- Our vision of chat rooms differs from other developers
  - Misuse of P2P API in our classes
  - Narrow usability, only .Net will be able to use
  - Over encapsulation
  - Use of managed code
- Team 6: Microsoft P2P
- 4-38

- ### Project Schedule
1. Start learning P2P API
    - a) Goal: PNRP
    - b) Date: 1/27/06
  2. Learning C#
    - a) Goal: read tutorials
    - b) Date: 1/29/06
  3. Learning C#
    - a) Goal: write basic C# app
    - b) Date: 1/29/06
  4. Learning Graphing
    - a) Goal: understanding P2P graphing capabilities
    - b) Date: 2/1/06
- Team 6: Microsoft P2P
- 4-39

- ### Project Schedule
5. Start encapsulating P2P framework
    - a) Goal: Learn P/Invoke
    - b) Date: 2/5/06
  6. Complete graphing framework
    - a) Goal: Graphing classes made
    - b) Date: 2/8/06
  7. Complete chat class
    - a) Goal: Implement one of our chat classes
    - b) Date: 2/15/06
  8. Complete first prototype
    - a) Goal: Use chat class to make a prototype
    - b) Date: 2/20/06
- Team 6: Microsoft P2P
- 4-40

- ### Project Schedule
9. Continue developing library
    - a) Goal: Implement all chat classes
    - b) Date: 3/13/06
  10. Develop second prototype
    - a) Goal: Use chat classes to make chat room
    - b) Date: 3/15/06
  11. Testing of prototype
    - a) Goal: Try to break prototype and find bugs
    - b) Date: 3/29/06
  12. Documentation (video, user docs, tutorial)
    - a) Goal: Demonstrate tools to develop chat room
    - b) Date: 4/12/06
- Team 6: Microsoft P2P
- 4-41

## 6. & 7. Teams: Technical Specification / Schedule Eclipse Plug-in for Advanced Code Generation

Team 7: Motorola  
CSE 498, Collaborative Design

Jen Chiang  
Tom Lilly  
Ashlee McFarland  
Jon Napier

Department of Computer Science and Engineering  
Michigan State University  
Spring 2006

## S Project Overview

- Develop an Eclipse plug-in for creating, editing, and storing Unified Modeling Language (UML) models.
- Import third party software files representing UML models to create new UML models
  - Third party software does not separate presentation specific data from model specific data
  - This application stores the UML models as two separate files, one for presentation data and one for model data

Team 7: Motorola

4-43

## S Project Overview

- This plug-in is built on the Eclipse Rich Client Platform (RCP).
  - Framework for creating applications with a rich user experience
  - This allows for the application be run as a console and as a part of the Eclipse integrated development environment (IDE).
- This plug-in is the first step in the process of creating source code from a UML Model in the Eclipse IDE.

4-44

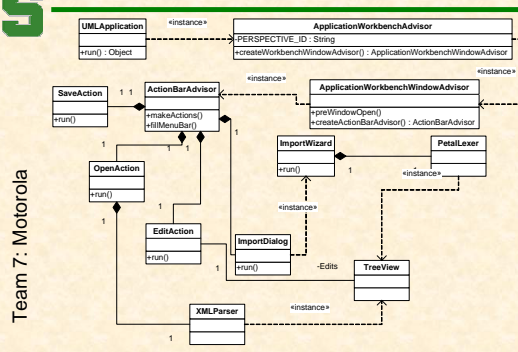
## S Architecture Components

- Hardware Platforms
  - Web Server (Windows 2003 running Apache)
  - Windows XP Professional
- Software Platforms / Technologies
  - Java
  - Antlr (ANother Tool for Language Recognition)
  - XML

Team 7: Motorola

4-45

## S Architecture Illustrated



Team 7: Motorola

4-46

## S Architecture Risks

- Finding and implementing the correct extension points from existing plug-ins.
- Finding documentation/examples for developing Eclipse plug-ins.
- Understanding Antlr in order to use a grammar file as a place to write code.
- Understanding the format of the third party UML model file.

Team 7: Motorola

4-47

## S Project Schedule

- We are following a modified version of Extreme Programming (XP).
- XP utilizes short cycle times and does not provide a complete schedule.
- A listing of the desired functionality is created for the project and a subset of this list is selected for a milestone.
- A new milestone with a new subset of functionality is set when the current one is completed.

Team 7: Motorola

4-48



## S Project Schedule

- Milestone 1:** (Due Tuesday, 1/31)
  - Create a plug-in with a perspective that contains three views (tree, editor, console) .
  - Implement the import functionality to parse a third party model file and create basic tree view hierarchy.
  - Add open and save shell functionality.
  - Be able to install plug-in(s) into console as well as IDE.
  - Add functionality to update the plug-in with new versions from a web server.
  - Create and use an update repository location on the web server.
  - Add functionality for altering the application's preferences.
  - Create Javadocs for application documentation.

4-49

  
**6. & 7. Teams:**  
**Technical Specification / Schedule**  
**Web-Based Video Editor**  
  
**Team 8: TechSmith**  
 CSE 498, Collaborative Design  
  
Jared Riley  
 Aaron Clarke  
 Tan Nguyen  
 Mai Luong  
 Department of Computer Science and Engineering  
 Michigan State University  
 Spring 2006



## S Project Overview

- The user will be able to manipulate time codes in such a way that it will appear that they are editing and viewing a completely new video across the web.
- The embedded web UI will retrieve and load video time codes to jump around locations in various movie files on various servers.
- An end user will be able to select parts from various video clips that they like, sequence them, and in the end they will have a link to provide to others that will show the composite video.

Team 8: TechSmith

4-51

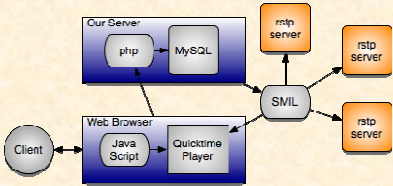
## S Architecture Components

- Hardware Platforms
  - 2 x86 computers (Server and Development PCs)
- Software Platforms / Technologies
  - Server**
    - Windows Server 2003 – Server
    - MySQL – Database
    - Darwin Streaming Server 5 - for hosting streaming media
    - PHP 4 for CGI applications
    - SMIL - XML variant used for creating multimedia presentations
    - Microsoft IIS - Project's web server
  - User**
    - Window XP Professional - development computer, can be any OS
    - Most Common Web Browsers (Firefox, IE)
    - JavaScript
    - QuickTime Player - Media-player to be embedded into web browsers

Team 8: TechSmith

4-52

## S Architecture Illustrated



Team 8: TechSmith

4-53

## S Architecture Risks

- Using JavaScript to query and control the QuickTime player, UI.
- Retrieve and load video time codes to jump around locations in various movie files on the server.
- Create interface between QuickTime and MySQL
- Generate SMIL file based on user inputs
- Learning advanced JavaScript & PHP

Team 8: TechSmith

4-54

## Project Schedule

Team 8: TechSmith

1. Embedded QuickTime Control
  - a) Goal: Play/stop/receive time code from QT
  - b) Date: 2/7
2. MySQL Store/Retrieve
  - a) Goal: Save/retrieve time code to/from server
  - b) Date:2/7
3. SMIL Creation
  - a) Goal: Generate a SMIL play list with PHP
  - b) Date:2/7
4. JavaScript Post
  - a) Goal: Create very basic interface between QT and MySQL
  - b) Date:2/13

4-55

## Project Schedule

Team 8: TechSmith

5. Custom SMIL
  - a) Goal: Create link between SMIL generated and MySQL data.
  - b) Date: 2/13
6. Sequence Edit
  - a) Goal: Rearrange order in which time codes were submitted
  - b) Date:2/20
7. Annotation Post
  - a) Goal: Save annotation data to MySQL
  - b) Date:3/7
8. SMIL with Annotation
  - a) Goal: Generate SMIL files with video and textual data
  - b) Date:3/14

4-56

## Project Schedule

Team 8: TechSmith

9. Graphic Timeline
  - a) Goal: Generate a graphical representation of videos timeline.
  - b) Date: 3/21
10. User Accounts
  - a) Goal: Multi-user capabilities
  - b) Date:3/30
11. Upload Videos
  - a) Goal: If possible, create interface for uploading own videos to server
  - b) Date:4/11
12. Voiceover
  - a) Goal: Add custom audio data to SMIL files
  - b) Date:4/18

4-57

MICHIGAN STATE UNIVERSITY

## 6. & 7. Teams: Technical Specification / Schedule Consultation Visit Reporting System

Team 9: **TWO MEN AND A TRUCK®**  
CSE 498, Collaborative Design

Pam Johnston  
Lauren Howard  
A.J. Orans

Department of Computer Science and Engineering  
Michigan State University

Spring 2006



## Project Overview

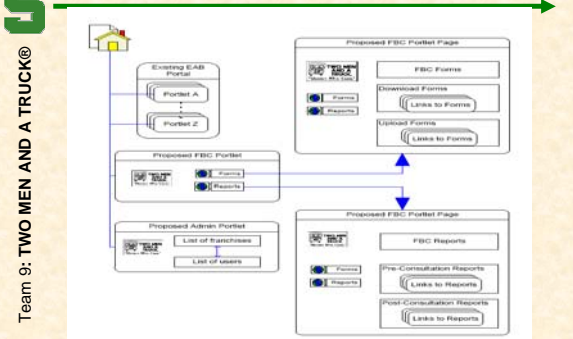
Team 9: **TWO MEN AND A TRUCK®**

- Streamline FBC franchise review process
- Electronic data entry / retrieval of franchise data
- Offline data entry utilizing InfoPath forms
- Database stored in SQL Server 2000
- Implementation of XML schema / Web Services
- Online portlet interface
  - Existing BEA Portal used for project management
  - Administrator associate FBC(s) with franchises
  - FBC download InfoPath form templates to tablet PC
  - FBC download reports with historical data – pull data
  - FBC upload completed forms – push data

4-59

## Overview Schematic

Team 9: **TWO MEN AND A TRUCK®**



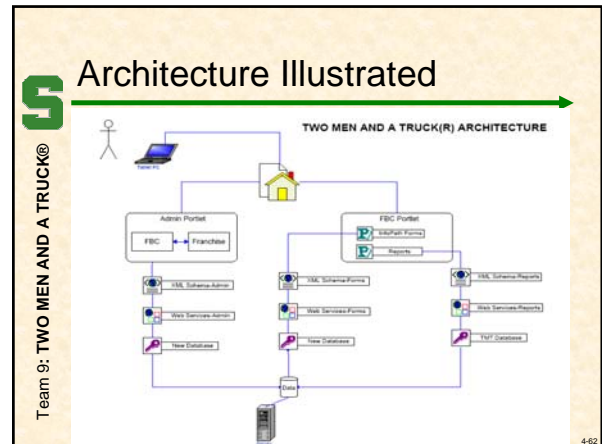
4-60

## Architecture Components

- Hardware Platforms
  - Tablet PC
  - PC and Servers
- Software Platforms / Technologies
  - Windows 2003 Server / IIS 6.0
  - Windows XP Pro
  - SQL Server 2000
  - BEA portal software, develop portlets with APIs
  - Visual Studio 2003 – ASP.NET (C#)
  - Microsoft Office – InfoPath 2003
  - XML Schema / Web Services

Team 9: TWO MEN AND A TRUCK®

4-61



## Architecture Risks

- Bind data in InfoPath XML to SQL database
  - How use XML schema and Web Services?
- Error checking for XML documents
  - How recover when incorrect data is entered?
- Scalability of design
  - Ease of adding data or new forms
- Security
  - How secure is data in front of corporate firewall using XML schema and web services?

Team 9: TWO MEN AND A TRUCK®

4-63

## Project Schedule

- 1) Create example form
  - a. Goal: Initial InfoPath form (no pre-population)
  - b. Date: January 30
- 2) Initial design
  - a. Goal: Detailed Technical Specification for review
  - b. Date: February 1
- 3) Create portlet UI
  - a. Goal: Create layout for portlet UI
  - b. Date: February 6
- 4) Create portlet
  - a. Goal: Implement basic functionality (no data)
  - b. Date: February 10

Team 9: TWO MEN AND A TRUCK®

4-64

## Project Schedule

- 5) InfoPath / SQL database connectivity
  - a. Goal: Data entry InfoPath form – push data
  - b. Date: February 11
- 6) SQL database / portlet connectivity
  - a. Goal: Implement administrative functionality
  - b. Date: February 17
- 7) Prototype
  - a. Goal: InfoPath form working within portlet
  - b. Date: February 20
- 8) InfoPath Error-checking
  - a. Goal: Error-checking for data entry with InfoPath
  - b. Date: February 27

Team 9: TWO MEN AND A TRUCK®

4-65

## Project Schedule

- 9) Database reporting
  - a. Goal: Ability to generate reports – pull data
  - b. Date: March 13
- 10) Refinements / Additions
  - a. Goal: Refine all major problem areas, add forms
  - b. Date: March 31
- 11) Testing
  - a. Goal: Test and debug system
  - b. Date: April 20
- 12) Final
  - a. Goal: Prepare final report / presentation
  - b. Date: April 24

Team 9: TWO MEN AND A TRUCK®

4-66



**MICHIGAN STATE UNIVERSITY**

**6. & 7. Teams:  
Technical Specification / Schedule  
Union Pacific**

Team 10: Union Pacific  
CSE 498, Collaborative Design

Rick Curtis  
Tom Mchizler  
Sonny Gupta  
Hogyeong Jeong

Department of Computer Science and Engineering  
Michigan State University

Spring 2006



## Project Overview

- **Wireless Locomotive Car Sensor Network**
- Replaces wired network.  
Currently a system of cabling runs through train car connecting multiple devices to Central PC (ARC)
- Multi-point to point network.  
New system will allow multiple devices to relay data to a single base station connected to the ARC.

Team 10: Union Pacific

4-68

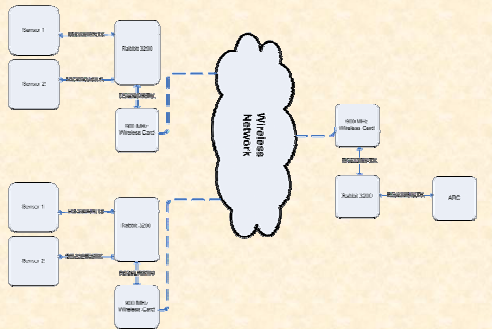
## Architecture Components

- **Hardware Platforms**
  - RabbitCore 3200 embedded processor
  - MaxStream 900 Mhz wireless device
  - Development boxes: Windows XP
- **Software Platforms / Technologies**
  - Dynamic C
  - VS .NET for writing test applications

Team 10: Union Pacific

4-69

## Architecture Illustrated



Team 10: Union Pacific

4-70

## Architecture Risks

- EM interference
- Wireless throughput
- Serial port throughput
- Developing a TDMA network protocol
- Processor limitations on rabbit boards
- Dynamic c limitations

Team 10: Union Pacific

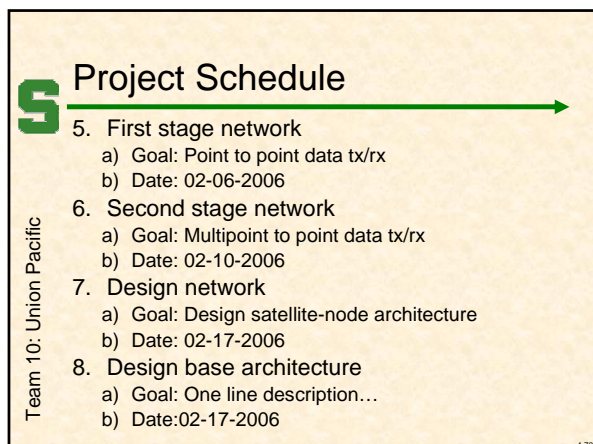
4-71

## Project Schedule

1. Rabbit power
  - a) Goal: Solder up rabbit power connections
  - b) Date: 01-31-2006
2. Rabbit serial connector
  - a) Goal: Solder up rabbit serial connections
  - b) Date: 01-31-2006
3. Rabbit tx application
  - a) Goal: Write a rabbit app that sends serial data
  - b) Date: 02-03-2006
4. Rabbit rx application
  - a) Goal: Write a rabbit app that receives serial data
  - b) Date: 02-03-2006

Team 10: Union Pacific

4-72



**S** Project Schedule

Team 10: Union Pacific

5. First stage network
  - a) Goal: Point to point data tx/rx
  - b) Date: 02-06-2006
6. Second stage network
  - a) Goal: Multipoint to point data tx/rx
  - b) Date: 02-10-2006
7. Design network
  - a) Goal: Design satellite-node architecture
  - b) Date: 02-17-2006
8. Design base architecture
  - a) Goal: One line description...
  - b) Date: 02-17-2006

4-73