Computerized Train Control System

Block Diagram

Shawn Lord

Christian Thompson

Kevin Sendra

Advisors:

Dr. Schertz

Dr. Malinowski

November 6, 2002
Overall Block Diagram:

**Client/Server** → **Controller Application** → **Local Controller** → **DCC Encoder**

**Client**

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Commands →</td>
<td>Web Cam Image →</td>
</tr>
<tr>
<td>Communication from Server →</td>
<td>Feedback to user →</td>
</tr>
<tr>
<td>Video Feed →</td>
<td>Communication to Server →</td>
</tr>
</tbody>
</table>

**Inputs:**

- User Commands
  - User input from java applet
- Communication from Server
  - Status and Error information from Server
- Video Feed

**Outputs:**

- Web Cam Image
  - Displayed on Screen
- Feedback to user
  - Information about the status is displayed for users
- Communication to Server
  - User Generated Commands sent to Server.
## Computerized Train Control System

<table>
<thead>
<tr>
<th>Server</th>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Feedback from Controller Application ➔</td>
<td>➔ Commands to Controller Application</td>
</tr>
<tr>
<td></td>
<td>Web Cam Image ➔</td>
<td>➔ Communication to Client</td>
</tr>
<tr>
<td></td>
<td>Communication from Client➔</td>
<td></td>
</tr>
</tbody>
</table>

**Inputs:**
- ➔ Feedback from Controller Application  
  Information on Layout Status and errors.
- ➔ Web Cam Image  
  Provided by a video input card and a camera.
- ➔ Client Communication  
  Commands sent from client connection

**Outputs:**
- ➥ Commands to Controller Application  
  Commands form User passed to Controller Application
- ➥ Communication to Client  
  Status, Information and Video Feed sent out to client
Client/Server Communication Flowchart:

1. Server Ready/Not ready for connection
   - Try to establish read and write connections with server
   - Return in use or error messages
   - Read Data about layout from server
     - If Error Occurs
     - Display error
   - Wait for command
     - Time out, close connection
     - Server sends error information
     - Client reads error and displays to user
     - Client disconnects
       - If Catastrophic Error
       - If Minor Error
   - Client sends a command that server receives
     - Server writes command to file for Controller Application
     - Server sends image to client
     - Client requests new image
     - Server sends error information
     - Client disconnects
       - If Catastrophic Error
       - If Minor Error
   - Server open for new connections
     - If Error Occurs
     - Display error
     - Client disconnects
       - If Catastrophic Error
       - If Minor Error
Controller Application

**Inputs:**
- Commands from Server → Commands to Local Controller(s)
- Status from Local Controller(s) → System Status to Server
- System Definition Files → System Definition Files

**Outputs:**
- Commands from Server
  - The Server will give commands to the Controller Application.
- Status from Local Controller
  - A protocol will be used so that the Command Application can poll the Local Controllers to check the current state of the sensors.
- System Definition Files
  - Text files which store information about the train layout, trains and system setup

**Inputs:**
- Commands from Server
- Status from Local Controller
- System Definition Files

**Outputs:**
- System Status
  - System errors and layout status sent to the Server.
- Output to Local Controller
  - Commands sent out to control the trains on the Local Controller’s segment of the layout. These will be sent over a serial network.
- System Definition Files
  - Text files which store information about the train layout, trains and system setup.
Controller Application Flowchart:

- **Startup** → **Poll Local Controller** → **Receive Status and write to File**

- **Start Engine** → **Receive Status and write to File** → **Repeat for all Locomotives**

- **Wait for Input** → **Recheck system status** → **Receive Status and write to File**

- **If Input is from Server** → **Check if command is valid** → **Send error message to Server**

- **If Input is from Local Computer** → **Send command to appropriate Local Controller**

- **When finished with all Locomotives** → **If no Input** → **If command is valid** → **Else**
Local Controller

**Inputs:**
- Controller Application Commands
- Sensors
- Handshake Signals from Decoder

**Outputs:**
- Data to Encoder
- Status Update to Controller Application
- Switch control

**Inputs:**
- Controller Application Commands
- Sensors
  - Block sensors to tell if there is an operating locomotive on a particular section of track.
- Handshake Signals from Decoder

**Outputs:**
- Status Update to Controller Application
- Data to encoder
  - Data is sent to the encoder module though a parallel connection.
- Switch Control
  - Ports which can be turned on or off by the Local Controller to control switches or other accessories.
Local Controller Flowchart:

Start

Receive Serial Command from Controller Application

My address

Not my address

Decode transmission

Command for status of layout

Send Sensor, switch, and train status on serial connection

Command to activate switch

Add Locomotive and command to queue

Command for Locomotive

New Locomotive

Old Locomotive

Overwrite command in queue
Encoder

Inputs:
Data from Local Controller

Outputs:
- DCC signal to tracks
- Handshake Signals to Local Controller

Inputs:
- Data from Local Controller
  Bytes of information from the Local Controller as well as Strobe and Packet Start bits which are placed into the input buffer.

Outputs:
- Handshake Signals to Local Controller
  Busy bit lets the Local Controller know that data has been received and is input buffer is full
- DCC signal to tracks
  Packets of information compliant with the NMRA standards for DCC are continuously sent to the tracks.

Software Flowchart: