CSULB Marine Biology Department Software Project

Requirements Analysis (Preliminary rev 3)

This document is subject to change. The latest version may be found at https://github.com/ashawnbandy/cecs491/tree/master/docs

10/9/2012

Contents

INTRODUCTION

Purpose of the System

The CSULB Marine Biology department collects data from marine life that has been tagged with an acoustic transmitter. Data will be collected by a receiver located off of Manhattan Beach Pier (MBP), which will record the ID number of tags, associated sensor data, date, and time of detection. The receiver in turn will be connected to a computer through a serial port connection. The purpose of the system will be to interface with this computer remotely in order to control the receiver and receive data from it.

Objectives

1. Ability to connect remotely to the computer managing the receiver
2. Ability to control the receiver remotely through that connection.
3. Ability to ~~optionally~~ stream real-time data from the receiver.
4. Send~~ing~~ ~~an~~ email alerts when a detection meeting user-defined criteria is ~~recorded~~observed.
5. Archive recorded data and recording metadata.

Definitions

1. "The system" will refer to the software being created by this project, and not the firmware on the receiver equipment.
2. Software connecting to the remote sites (e.g. Manhattan Beach Pier) will be referred to as the "server" or "backend".
3. Software running on or transferred from csulbsharklab.com will be referred to as the "client" or "front-end"

The roles of "User" and "Administrator" is defined by the csulbsharklab.com software and documentation.

4.

Real-Time-Mode (RTM) refers to the presentation of data as it comes into the server from the receiver and may or not utilize Real Time Mode 0 on the receiver hardware.~~References~~

~~1. "VR2C wired acoustic receiver", submitted on 2012-09-17~~

5.

Formatted: Numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 1" + Indent at: 1.25"

Formatted: List Paragraph, Numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 1" + Indent at: 1.25", Border: Top: (No border), Bottom: (No border), Left: (No border), Right: (No border), Between : (No border), Bar : (No border)

Formatted: Numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 1" + Indent at: 1.25"

FUNCTIONAL REQUIREMENTS

1. Connect to receivers located remotely (e.g. MBP)
2. Start and stop recording data from receiver to server
3. Provide connection to receiver through console
4. Parse and aggregate data sent from receiver
5. Translate data received to SQL commands
6. Send generated SQL commands to csulbsharklab.com automatically for later retrieval by end-users.
7. Email alerts sent out when the system detects user defined parameter

NON-FUNCTIONAL REQUIREMENTS

Usability

- The application will minimize network configuration.
- Additional receiver nodes with up to *N* receivers will also require minimal configuration by a remote operator.
- The front-end user interface will follow design practices in the csulbsharkalb.com. Any server user interface will follow familiar design practices.

Reliability

- The server should be continuously available. To this end, the system should detect critical faults and reset without end-user administration.
- Non-critical faults will be either logged or reported to a console.

Safety

There are no known safety requirements.

Security

- Access to the client will be handled by the existing csulbsharklab.com user validation system.
- Data identified as sensitive will be encrypted when transmitted over open networks (e.g. the internet, the CSULB network, etc.)

Performance

- Commands to receivers and their effects should be sent and received in near-real-time.
- ~~Remote servers should enter ready status within *N* minutes of a cold start.~~

Supportability

- Sufficient documentation will be provided to the customer to allow for future bug fixes by a third-party.
- The design will be modular and make use of simple declarative language control files (i.e. json) to allow for future changes or additions to the system.

Implementation

- The server software will be written in C# for deployment on Microsoft Windows machines.
- The server software that interfaces with the hardware will be written as part of this project.

- The server module connecting to the MySQL database will be a standard library from Microsoft and/or Oracle (publisher of MySQL).
- Configuration files will be implemented using a simple declarative language such as JSON (javascript object notation).
- Network communication between remote sites and the server will be handled by serial-over-ethernet hardware devices.

Interface

- The system will interface with the firmware (current at time of implementation) on each receiver.
- The system will generate data consistent with existing output formats (SQL).

Packaging

- Server software will be installed at a site on the California State University of Long Beach campus by a member of the team.
- All software will also be packaged in a manner that facilitates additional installations.

USE CASES

~~m~~

~~**Name**~~	~~Change to real-time mode~~
~~**Actor(s)**~~	~~Administrator~~
~~**Pre-conditions**~~	~~The user is connected to the server.~~
~~**Flow-of-Control**~~	~~Network connection to server~~ ~~1. Administrator or User has logged in with valid credentials.The user selects real-time mode from the UI.~~ ~~2. The system switches to real-time mode.~~
~~**Post-conditions**~~	~~The system is set to real-time mode.~~

M

Name	Connect to VR2C Receiver (Receiver Life-Cycle)
Actor(s)	Serial Port (.NET hardware abstraction)
Pre-conditions	~~None~~A running service manager.
Flow-of-Control	1. New serial port is discovered. 2. VR2C receiver is configured 3. Receiver software sends commands and receives messages from the hardware until directed to stop.
Post-conditions	Receiver software disconnects from the hardware and is removed from the system.

Y

Name	Setup email alert(s)
Actor(s)	Administrator
Pre-conditions	1. Network connection to server. 2. Administrator has logged with valid credentials ~~None~~
Flow-of-Control	1. The user selects Settings → Email Alert ~~click alert setting button on the main window then it opens the email setting window separately.~~from menu. 2. The user enters an email address on the text box and clicks add button to enter the email on the email list.

Formatted: Heading 1

Formatted: Font: (Default) +Body (Century Gothic), 12 pt, No underline, Font color: Text 2

Formatted: Heading 3, Space After: 0 pt, Line spacing: single

Formatted: Font: +Body (Century Gothic), 10 pt, No underline

Formatted: Font: 10 pt

Formatted: Font: +Body (Century Gothic), 10 pt, No underline

Formatted: Font: 10 pt

Formatted: Font: +Body (Century Gothic), 10 pt, No underline

Formatted: Font: 10 pt

Formatted: Font: +Body (Century Gothic), 10 pt, No underline

Formatted: Font: 10 pt

Formatted: Font: +Body (Century Gothic), 10 pt, No underline

Formatted: Font: 10 pt

Formatted ...

Formatted ...

Formatted: Font: 10 pt

Formatted ...

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: No underline

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted: Font: 10 pt

Formatted ...

Formatted: Font: 10 pt, Not Bold

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted ...

	3. The user setup filter for the email alert(s) based on location/receiver/specific tag/time etc. 4. The user clicks save to save the email alert setup. 5. The user clicks red X button on the right side upper corner to close the email setting window. ~~1. The user specifies a server to connect to and credentials to connect with.~~ ~~2. The client connects to the server and is authenticated, or denied.~~
Post-conditions	The server machine sends out email alert(s) to the email addresses on the email list when the trigger condition occurs.~~The user is connected to the server, or an error is returned.~~

	Y
Name	Start/stop Receiver Hardware
Actor(s)	Administrator
Pre-conditions	1. An open connection to the receiver hardware to be stopped or started.
Flow-of-Control	1. User chooses to start or stop the receiver on the server user interface. 2. A valid command is sent from the receiver software to VR2C hardware. 3. The hardware begins to send new detections if started, or is placed in storage mode if stopped.
Post-conditions	1. The VR2C receiver has been started or stopped as directed by the user.

	D
Name	Access/query/~~save~~ recorded "real time" data
Actor(s)	Administrator, User
Pre-conditions	1. Network connection to csulbsharklab.com

Formatted: Font: +Body (Century Gothic), 10 pt, No underline

Formatted: Font: +Body (Century Gothic), 10 pt

Formatted: Font: +Body (Century Gothic), 10 pt, No underline

Formatted: Font: +Body (Century Gothic), 10 pt, No underline

Formatted: Font: +Body (Century Gothic), 10 pt

Formatted: Font: +Body (Century Gothic), 10 pt, No underline

Formatted: Font: +Body (Century Gothic), 10 pt, No underline

Formatted: Font: +Body (Century Gothic), 10 pt

Formatted: No underline, Font color: Black

Formatted: Normal, Space After: 0 pt, Line spacing: single, No bullets or numbering

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: +Body (Century Gothic), 10 pt, No underline

Formatted: Indent: Left: 0.25", Space Before: Auto, After: Auto, Line spacing: single, Font Alignment: Baseline

Formatted: Font: 10 pt, No underline

Formatted: Font: +Body (Century Gothic), 10 pt, No underline

Formatted: Font: 10 pt

Formatted ...

Formatted ...

Formatted ...

Formatted ...

Formatted: Font: Not Bold

Formatted: Font: 10 pt, No underline

Formatted: Font: 10 pt

Formatted ...

Formatted: Font: 10 pt, Font color: Black

Formatted ...

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Line spacing: single

Formatted: Font: 10 pt, No underline

Formatted: Font: 10 pt

Formatted ...

Formatted: Font: 10 pt

	~~1.~~2. Csulbsharklab.com is running. 3. User has logged into csulbsharklab.com with valid credentials.
Flow-of-Control	~~User clicks "Visitor Query" link.~~ ~~.~~ ~~User selects "save data" button~~ ~~6. User selects directory to save data on local computer.~~
Post-conditions	~~If user chooses to save their queried data, data is saved into user directory~~Query results sent to the browser.

Formatted: Line spacing: single, Outline numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 0.25" + Tab after: 0.5" + Indent at: 0.5"

Formatted: Font: Font color: Black

Formatted: List Paragraph, Space After: 0 pt, Line spacing: single, Numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 0.25" + Indent at: 0.5"

Formatted: List Paragraph, Space After: 0 pt, Line spacing: single

Formatted: List Paragraph, Indent: Left: 0", Space Before: 0 pt, After: 0 pt, Line spacing: single, Font Alignment: Auto

Formatted: Font: 10 pt

Formatted: Line spacing: single

	D
Name	Store recorded data
Actor(s)	Hardware Receiver, MySQL database
Pre-conditions	1. MySQL database server must be running and available to the server software.
Flow-of-Control	1. New data arrives from receiver hardware and is properly decoded. 2. When appropriate SQL statements are generated from raw data and sent to remote MySQL database server.
Post-conditions	Data has been stored in remote database.

Formatted: Line spacing: single, Outline numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 0.25" + Tab after: 0.5" + Indent at: 0.5"

Formatted: List Paragraph, Space After: 0 pt, Line spacing: single, Numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 0.25" + Indent at: 0.5"

Formatted: Space After: 0 pt, Line spacing: single

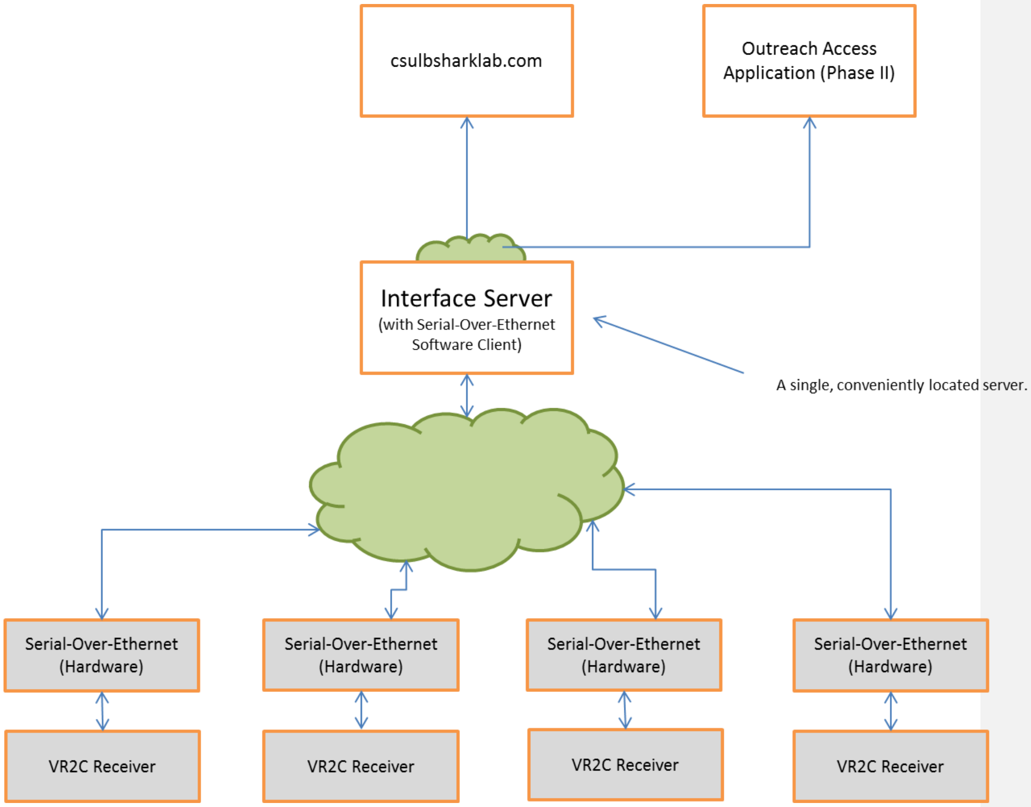
	D
~~Name~~	~~Begin Data Streaming~~
~~Actor(s)~~	~~Administrator, User~~
~~Pre-conditions~~	~~Network connection to server~~ ~~Administrator or User has logged in with valid credentials.~~
~~Flow-of-Control~~	~~User selects "real time stream" button.~~ ~~System opens the real time streaming user interface.~~ ~~System periodically updates panel with new data from server.~~
~~Post-conditions~~	~~System will update panel with new data as it becomes available.~~

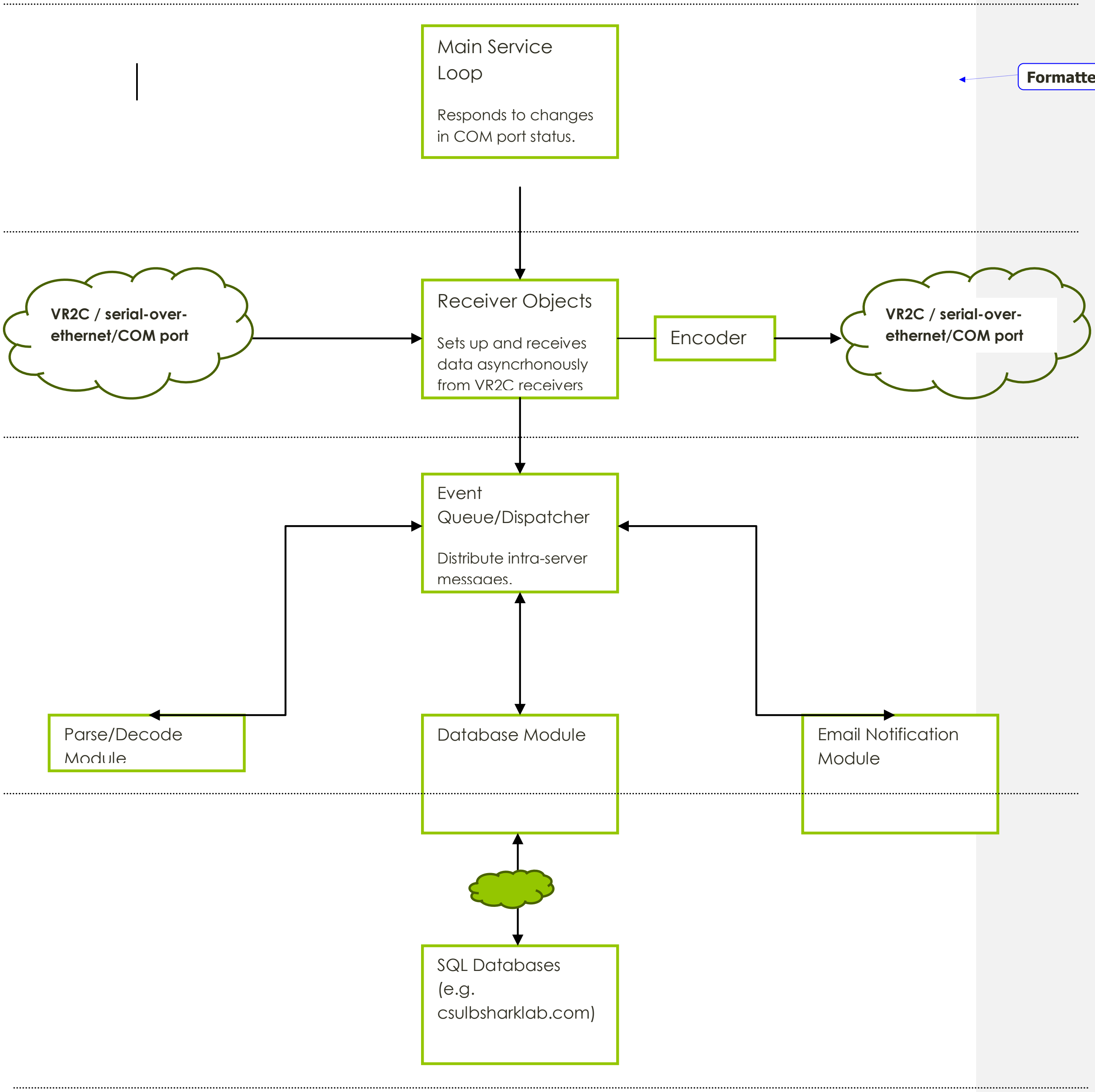
Formatted: Line spacing: single, Outline numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 0.25" + Tab after: 0.5" + Indent at: 0.5"

Formatted: List Paragraph, Space After: 0 pt, Line spacing: single, Outline numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 0.25" + Indent at: 0.5"

Formatted: Heading 1

APPENDIX 1 – SYSTEM OVERVIEW DIAGRAMS (UPDATED 10/09/2012)


csulbsharklab.com
Outreach Access Application (Phase II)
Interface Server
(with Serial-Over-Ethernet Software Client)
A single, conveniently located server.
Serial-Over-Ethernet (Hardware)
Serial-Over-Ethernet (Hardware)
Serial-Over-Ethernet (Hardware)
Serial-Over-Ethernet (Hardware)
VR2C Receiver
VR2C Receiver
VR2C Receiver
VR2C Receiver

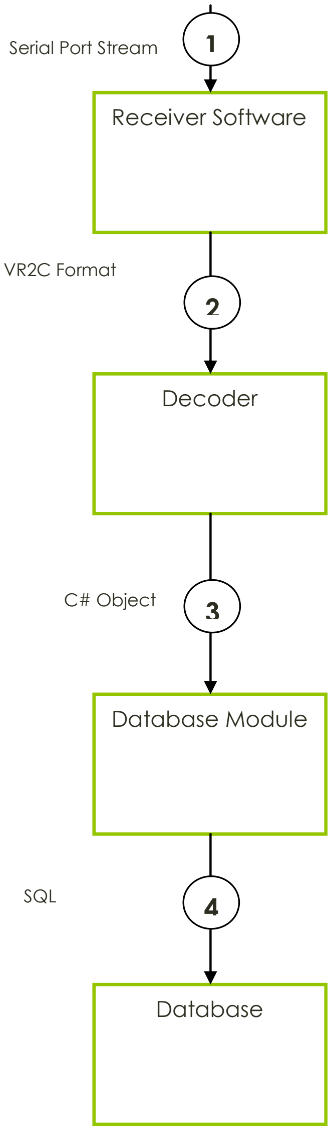

Main Service Loop
Responds to changes in COM port status.
VR2C / serial-over-ethernet/COM port
Receiver Objects
Sets up and receives data asyncrhonously from VR2C receivers
Encoder
VR2C / serial-over-ethernet/COM port
Event Queue/Dispatcher
Distribute intra-server messages.
Parse/Decode Module
Database Module
Email Notification Module
SQL Databases (e.g. csulbsharklab.com)

Formatted: Heading 2

Note: Objects between common dashed lines represent serial executions (common thread of execution).

Appendix 2 – Data Flow Diagrams (Added 10/09/2012)

"Data Browser" Motif


Serial Port Stream
1
Receiver Software
VR2C Format
2
Decoder
C# Object
3
Database Module
SQL
4
Database