From Students…
…to Professionals

Project Plan

Dynamic Spectrum Control for Mobile Platforms

The Capstone Experience

Team Raytheon

Will Bonner
Matt Bowser
Srinivasa Settaluri
Jimmy Voss

Department of Computer Science and Engineering
Michigan State University
Spring 2011
Project Overview

• Sample surrounding radio frequency spectrum
• Display data on graph, offer recommendations to user
• Provide interface to specify policies used in making recommendations
• Send results to EZ-Connect API to tune military radios to recommended frequencies
Functional Specifications

• Clearly displays RFS data
• Provides interface for creating and selecting from a set of policies
• Saves policies from time of use
• Provides user with a set of recommendations (single response to requesting application)
Design Specifications

• RFS data graph is scalable, displays multiple recommendations simultaneously (with inactive rec. at ~50% alpha)
• User can specify how many recommendations are selectable
• Controls to edit/delete policies are a context sensitive menu (long touch at menu)
Screen Mockups
Screen Mockups

[Diagram showing a user interface with options such as 'Currently Active Policy', 'Current recommendation', 'Rec 1', 'Rec 2', 'Rec 3', 'Change Policy', 'Pick from Recommendations', and buttons to 'View the first recommendation', 'View the second recommendation', 'View the third recommendation'.]
Technical Specifications

• Alpha version developed on Android, Beta includes port to Laptop (OS agnostic)
• Primary algorithm written in C, implemented in app with Android NDK
• Policy data stored in SQLite database local to device
• Set of recommendations displayed dynamically
System Architecture

[Diagram showing system architecture with components labeled as RF Sampling Hardware, 3rd party app, Army Radio, TCP data, direct interface, TCP rec, DCS Software, Pick Policy, Algorithm, Display, Policy, Rec]
System Components

• Hardware Platforms
  ▪ RF sampling hardware
  ▪ Android/Laptop interface
  ▪ Radio

• Software Platforms / Technologies
  ▪ Android OS, QT
  ▪ C, Java, SQLite
  ▪ TCP/IP
Testing

- Interface/data testing: Android Emulator
- Algorithm efficiency/accuracy testing: Command line
- First round device testing: Visit to Raytheon (pre-Alpha presentation)
- Laptop port testing: Lab machine/personal laptops
- Full functionality testing: Second visit to Raytheon (pre-Beta presentation)
Risks

• EZ-Connect
• Real Radio Data
• Connection to someone else would be impossible
• Executing code with the NDK
• Srinivasa is not a US citizen