System Requirements Specification for : Patient Service System

Rohit Saraf	Ashish Mathew	Vivek Madan	Pritish Kamath
08005040	08D10035	08005035	08005004

April 13, 2011

under the guidance of
Prof. Kavi Arya

Embedded and Real-Time Systems Laboratory
Department of Computer Science and Engineering
Indian Institute of Technology
Bombay

Contents

1 Introduction

The main idea of this project is to develop an automated system for assisting patients in hospitals.

1.1 Definitions, Acronyms and Abbreviations

- **Centralized Server :** Receives signals from patients in the hospital, runs a scheduling algorithm and signals the robotic attendents or sends messages to doctors.
- **Robotic Attendent :** Receives commands from the centralized server and proceeds to serve the patients.

1.2 References

- NeX robotics
- AVR on Ubuntu
- ATmega2560 datasheet
- FireBird V Software Manual
- FireBird V Hardware Manual

2 Overall Description

2.1 Product Perspective

Patients in hospitals often require 24-hour attention for their basic needs. However attendants can't be available for patients at all times. Also it may often happen that because of hectic hospital schedules, attendents may forget to give medicines to patients on time. This induces a need for an automatic attendent who could possibly replace human attendants for almost all routine tasks, thus providing more reliability and reducing a tremendous amount of work-load on nursing stations.

2.2 Product Functions

The automatic attendent can serve patients' requests such as supplying water or patient-specific medicines and it would be summoned using an easy to use interface by the patients. Possibly the hospital buzzer interface can be integrated with the bot-summoning controller.

2.3 User Characteristics

The user (patients in hospitals) should be physically able to operate a controller (such as a mobile phone).

2.4 Constraints

The attendent service should be real-time. It should serve all patients in a fair manner (i.e. any patient request should have a bounded wait)

2.5 Assumptions and Dependencies

- There should be a clearly demarcated path for the attendant to follow.
- The wireless connectivity between the centralized server and the attendant-bot should be flawless.
- It is assumed that the pharmacy will be a manned dispensing station from which the attendant can collect medicines using a fixed protocol.
- Also the attendant should be have a port for charging where it would need human assistance.

3 Details

3.1 Functionality

- Central Server
  - Patient requests should be communicated to the centralized server through a wireless medium.
  - The centralized server would send attendants to serve the patients' requests or communicate to doctors about the urgency of the situation (through SMS or otherwise), whichever applicable.
  - It should run a scheduling algorithm to manage the robotic attendants so as to serve the patients in a least possible time.
- Robotic Attendants
  - The robot should follow a fixed path indicated by white lines.
  - It should stop in presence of an obstruction and alarm the guards about its prolonged obstructed movement.
  - It should be able to serve multiple patients, on request.

3.2 Supportability

- The code should be written in a modular fashion.
- Interfaces of each module should be properly defined.
- Standard naming conventions should be followed for variables and functions.

- In addition these variable and function names should be self explanatory.
- The code should be well commented.

3.3 Design Constraints

- Fairness should be ensured while answering all patient requests, that is, no request should go unanswered beyond a bounded time interval.
- The robotic-attendants should satisfy Asimov's three laws of robotics.

3.4 On-line User Documentation and Help System Requirements

- Detailed Documentation
- Project Final Report

3.5 Interfaces

The system requires the following interfaces for its functionality.

3.5.1 User Interfaces

- An easy to use interface which communicates wirelessly with the centralized server (such as a smart phone).

3.5.2 Hardware Interfaces

- ZigBee interfaces at robots, user interface and the central server.

3.5.3 Software Interfaces

- AVR-dude programmer
- AVR-gcc compiler
- AVR-libc library

3.5.4 Communications Interfaces

- ZigBee