Cody Adams

CSC-383 Fall 2008

Assignment #8

Functional Requirements

5.1. Robot Client/Server

5.1.1. Distributed Model1

The system shall allow distributed computing, where the robot may be local (on the same computer) or remote (via a network connection). One possible implementation is to use a proxy object – the proxy object (client) would communicate with the actual object (server) via the network. This is similar to the current mrcApiRemote (proxy) and mrcApiLocal classes. Another possibility is to use CORBA.

1 This section is sufficiently specific, complete, correct, understandable, and verifiable.

5.1.2. Robot Definitions2

The system shall support the concept of a “composite robot”, i.e., robot that is created from an ordered composition of other robots. For example, the 3 base translations and the RCM of the Steady Hand robot can be considered as two separate robots, as well as one combined robot.

2 This section is sufficiently specific, complete, correct, understandable, and verifiable.

5.1.3. Position/Velocity Query Commands3

The API shall provide several position/velocity query commands that return the robot’s position/velocity in joint coordinates or as a frame (a specific coordinate-based reference point represented by a defined data structure). There may need to be more than one type of frame query command – for example, one to return the frame corresponding to the RCM point and one to return the frame corresponding to the tool (e.g., needle tip). Note that intermediate frames can be obtained by querying the position/velocity of a component robot (for example, to get the frame corresponding to the end of the 3 base translations). There will also be different position/velocity query commands corresponding to positions/velocities computed from the joint set points, the primary sensor feedback or the redundant sensor feedback (if present).

3 5.1.3 Frame was not properly defined, I inserted a definition for frame.

5.1.4. External Sensor Query Commands4

The API shall provide query commands for external sensors, such as force sensors, that are configured with the robot (and are therefore read by the robot software). This includes:

Steady Hand pressure detection sensors

Accelerometers for Relational Positioning Data

Velocity detection sensors

4 The word *any* was not verifiable, and thus omitted in this revised version. Sensors should also be defined her, I have attempt to demonstrate a few.

5.1.5. External Sensor Update Commands5

The API shall provide an interface with which Users can interact in real time to typical sensors such as joysticks.

5 This section violated the specification’s domain, by requiring an undefined amount of external sensors. I have completely reworded this section.

5.1.6. Status and Error Query Commands6

The API shall provide query commands for the overall system status/error as well as for the status/error of individual joints. See the Real Time Error Handling section for additional specifications.

6 The Real Time Error Handling section should not be omitted here. I did not re-invent the wheel, those specs should simply be inserted and re-evaluated as a whole in this new context.

5.1.7. System Configuration Commands7

The API shall include commands that allow a robot system to be configured from a configuration file. There shall also be a command to query a robot for its configuration data. Configuration data shall include the following items:

• Robot name and type

• External hardware definitions (e.g., board types)

• Kinematic parameters, including joint zero offsets

• Joint position, velocity and acceleration limits

• Servo control parameters

• Safety thresholds (e.g., max. tracking error)

• Sensor definitions, including conversion factors

7 This section is sufficiently specific, complete, correct, understandable, and verifiable.