Store Controller Service Requirements

Author: Eric Gieseke

Date: 10/10/2019

Introduction

This document provides the requirements for the Store 24x7 Controller Service.

Overview

The Store Controller Service is responsible for monitoring the state of the sensors and appliances within the store. In addition, the Controller Service is able to generate actions to control the appliances based on rules, in response to status updates from the sensors and appliances.

Sensors are able to collect and share data. Like Sensors, Appliances can collect and share data. However, appliances can also be controlled. Please refer to the Store Model Service Requirements document for more information.

Store Controller Requirements

This section defines the requirements for the Store Controller Service.

The Store Controller Service should support the following functions:

- Monitor Sensors and Appliances for status updates.
- Apply rules that respond to the status updates from sensors and appliances and generate actions.
- Sensor input includes voice commands received via the microphones.
- In response to actions, generate and send control messages to Appliances.
- All transactions are performed using the Blockchain Ledger using the Unit currency.

The Store Controller Service should use the interface of the Store Model Service to monitor the status of each of the IoT devices installed within the houses. In response to inputs, the Controller Service will use rules to invoke actions. The actions will be executed through the appliance controls.

All rule execution and resulting actions should be logged.

Design Input:

- Follow the modularity specified in Store 24x7 System Architecture document.
- Apply the Command Pattern to implement the Actions performed by the Store Controller Service.
- Apply the Observer Pattern to allow the Store Controller to “listen” for events emitted by the Sensors of the Model Service.
- Use the Ledger Service (from assignment 1) to check account balances and submit transactions for checkout.

Sensor, Stimulus, Rule, Action

The following table defines the behavior for the Controller Service. The Controller Service will monitor all sensors and appliances for each of the houses and rooms. For each stimulus, apply the appropriate action.

Name	Sensor or Appliance	Stimulus (within the context of a store <store>)	Action
Emergency	Camera	emergency <emergency> in <aisle> Where <emergency> is one of: fire flood earthquake armed intruder	action for <store> 1. Open all turnstiles 2. announce: "There is a <emergency> in <aisle>, please leave <store> immediately" 3. Robot 1: "address <emergency> in <aisle>" 4. remaining robots: "Assist customers leaving the <store>"
Basket Event	Camera	Customer <customer> (adds|removes) <product> from <aisle:shelf>.	1. Add/remove the product to/from <customer> basket 2. Remove/Add product from <aisle:shelf> 3. robot: perform task restock for <aisle:shelf> and

			<product>
Cleaning Event	Camera	<product> on floor <store:aisle>	Robot: "clean up <product> in <aisle>"
Broken glass	Microphone	sound of breaking glass in <aisle>	Robot: "clean up broken glass in <aisle>"
Missing person	Microphone	can you help me find <customer name>	locate customer <name> speaker: <name> is in aisle <aisle>
Customer Seen	Camera	Customer enter <aisle>	Update customer location <aisle>
Fetch Product	Microphone	<customer> says: Please get me <number> of <product>	robot command: fetch <number> of <product> from <aisle> and <shelf> and bring to customer <customer> in aisle <customer_location>
Check Account Balance	Microphone	Customer <customer> says "What is the total basket value?"	1. compute the value of items in the basket 2. check account balance 3. speaker: "total value of basket items is <value> which is (more | less) than you account balance of <balance>
Assist Customer to car	Turnstile	checkout: total weight of products in basket exceeds 10 lbs	request robot to assist customer <customer> to car
Checkout	Turnstile	customer <customer> approaches turnstile	1. Identify customer <customer> 2. compute the total cost of items in the basket 3. create transaction 4. submit the transaction to blockchain 5. open turnstile 6. goodbye message :"goodbye <customer_name>, thanks for shopping at <store_name>!"

Enter Store	Turnstile	<customer> waiting to enter at the turnstile <turnstile>	1. Lookup Customer<customer> 2. Check for positive account balance 3. Assign Customer a basket 4. open turnstile <turnstile> 5. welcome message "Hello <customer_name>, welcome to <store name>!"

Since this is a new service, feel free to experiment with other types of Rules.

Testing

Continue to use the Store Model Service Command Langauge to test your system.