**CSCI-E97 Software Design: Principles, Models, and Patterns**

**Assignment 3 - Smart City Controller**

# Results Document

Date: 10/26/2020

Author: Samer Akash

o **Did creating the design help make the implementation easier?**

Yes. One the design was completed the implementation was easier specially after building the class diagram using Astah.

○ **How could the design have been better, clearer, or made the implementation easier?**

I think the design could be better if I was able to see the requirements for the rest the system components at once to have an idea of the big picture.

I will also think about redesigning Rules mapping instead of hardcoding the rules as part of commands. Encapsulating rules on their own will make configuring the smart city easier.

**○ Describe implementation changes that you made to your design and how they continue to support the requirements?**

- eventBroker class, which is the object that contains the message that will be sent to observers.

- events history list of all eventsBroker messages that’s serves as an events logger

- added smartCityUtils utility class to the model service package. This utility class include functions like find nearest robot and distance calculator.

- the logic of execute() function in the Command pattern implementation, changed to encapsulate the rule and command in one object. A better design will be to encapsulate rules separately.

- The assumption that devices front end code will send proper event syntax.

- To simplify commands implementation, the location of the device triggering the event is the same as the event location.

- Admin use case will require implementing admin portal for configuring rules of commands.

- Command line processor changed to include ledger commands lines.

- The constructor of ModelService is adding Controller as an observer

**○ Did the design review help improve your design?**

The reviewer helped me to identify couple of missed requirements and pointed out some changes needed regarding the class diagram.

**○ Comments about your design from your peer design review partners?**

**From Gustavo Varo:**

* Review the use case diagram. the controller is not an actor.
* Use case diagram was missing ledger service.
* show the the command\_line utility class in the diagram. (its covered as part of the Model service design class diagram)
* add auth\_token to all public methods. (its covered as part of the Model service design class diagram)
* Logger for events and commands is missing.
* Sequence Diagram was missing the ledger service.

**From Tanner Marsh:**

Tanner was not available for the review process. We have tried to delay the review meeting as a help for him, but his design for review. Communications with him was hard.

**○ Comments provided by you for each of your peer design review partners**

**To Gustavo:**

- Add controller exception class to capture exceptions in the controller service.

- Performance issues are expected if the message that is notifying observers is the model service itself.

- Add auth auth\_token to public methods in the public interface.

- Add use case diagram composition and aggregation wherever is applicable.

- Add ledger service interface to the class diagram.

- Sequence diagram was diagram was little confusing.

**To Tanner:**

I was not able to receive the design from Tanner. He didn’t join our meeting and he didn’t reach out to us for a review.