

Results document

Did the design document make the implementation easier?

I definitely think the design document made the implementation easier. I think the sequence diagrams did a good job at helping me better understand the command and observer pattern as well as the overall design of the Smart City Controller service.

How could the design have been better?

I think the design could have been better if we had a more narrowly designed interface. Rather than parsing out speech for things like bus route and movie time I think we could have implemented a more defined interface like imaginary buttons, where we would know exactly what the user wants (based on their selection) rather than expecting them to say some exact phrase.

Did the design review help you improve your design?

I think it did it helped to see how other people were implementing the tow design patterns in order to improve on my own.

How did you find the integration of the components?

I thought they integrated pretty well although there seemed to be some incompatibility between the Controller Service and Ledger Service design regarding the blocksize specifically and how accounts aren't added to the Ledger until a block is complete.

Your comment on your peers designs

Our peer reviews were in a back and forth dialog and can be accessed here, please let me know if you cant access them. I've listed here the largest/starting comment of the peer review threads though just in case.

<https://canvas.harvard.edu/groups/92576>

“Good job on the design document. Your implementation is really similar to what I was thinking too but that makes sense since we're all required to use the command and observer patterns. Is there any particular reason you choose to just have one observer on the Smart City Service Model as opposed to some thing like an observer for each type of device?

Also, if you were to extend you sequence diagram section to include a diagrams for an event that uses the ledger then that would add another step to the sequence diagram right? Like a step in the end, from command, that calls out to the ledger service? Would it be safe to say that most of these events would result in the command executed at least calling back to the Model Service?

I was considering, in my design document, adding a sequence diagram that includes one of the more complicated calls that reaches out to both the ledger service and back to the model service. I'm wondering if you've considered that as well? or maybe i'm misunderstanding something about how sequence diagrams are supposed to be laid out?“ – Brett to Brain

“Your design document looks good, just a few things I had questions or comments on. I do remember there being some discussion about the built in observer pattern tools in Java but I don't recall if Eric ever mentioned it was okay to use those or if we had to implement our own. I implemented my own. but i thought it was worth mentioning. Its totally possible that he said its cool to use the ones provided by Java and I just missed that part of the lecture or something. It looks like your design uses a SmartCitySimulator and SmartCityModelService class to sort of combine in to the SmartCityModelService class from assignment 2 which looks cool. From your design it looks like the concrete commands dont have references to either of these classes? when a command needs something from the model service, will it just be going through the Controller class to execute that action on the model service?” – Brett to Anthony

Comments from your peers design review partners

Our peer reviews were in a back and forth dialog and can be accessed here, please let me know if you cant access them. I've listed here the largest/starting comment of the peer review threads though just in case.

<https://canvas.harvard.edu/groups/92576>

“Some comments:

- In the use cases section, should the use case “Detect Events” be a Model Service actor event, or a Controller Service actor event? The description seems to lean more towards it being a Model Service actor event.
- The class diagram appears to describe the classes that will need to be implemented to successfully complete the assignment.
- In the included sequence diagram, I believe that the diagram shows the notify() method in the model service ends and that the update() method is called afterwards. Is this you intention? I believe that you probably meant that the update() method is called inside the notify() method.
- I like your exception handling idea in that the controller will log exceptions and will not attempt to further execute the command, but otherwise continue handling the event stream.” - Brian to Brett

Changes to your design based on the peer design review or implementation

There weren't very many big design changes based on the reviews however I did redo my sequence diagram based on the review Brian provided.