Peer review Workshop 1

For domain model made by Frida Holmström

Author Monica Sundin



***As a developer would the model help you and why/why not?***

It gives some clues of what’s meant to be done but it also raises questions as it lacks some information. As a developer I read the names as function-names instead of attributes. Which on the other hand give me clues for what’s meant to happen with the class but it’s not correct naming of attributes which Larman describes to be “a logical data value of an object” [1, p158]

There are some correct conceptual classes as Member, Boat, Berth and Calendar from the real world domain according to the guidelines for “Thinking like a mapmaker” [1, 145] looking at the use-cases concerning member but it seems to lack information regarding use-cases concerning secretary. I couldn’t find another picture than this one so I assume this is it. Yet of course if it’s not meant that member also represents the role of a secretary?

***Do you think a domain expert (for example the Secretary) would understand the model why/why not?***

I’ll answer pretty much as in previous question. It gives some clues but as it’s not complete it also raises questions.

***What are the strong points of the model, what do you think is really good and why?***

It has a good start for the domain model with the conceptual classes included and the naming of the associations representing the real world using names from the domain.

***What are the weaknesses of the model, what do you think should be changed and why?***

I’ve started to answer this in previous questions. It seems to have function names instead of attributes. The naming suggests what’s to be done with each class which is not according to the rules for the domain model attributes.

To find the classes the literature suggest different methods, where one is that you may look for nouns in the use cases [1, p141] and to find the attributes look in the use-cases for what the requirements suggest or imply might be important [1, p158]. You don’t have to have attributes for all classes if they don’t add anything extra at this stage. For boat you can for example find size and boat-type as interesting attributes as the effect the membersfee to be paid. For Calendar titlt, stardate and enddate is mentioned.

I would also prefer calling calendar for calendarEvent to reflect the naming in the use-case as it’s a bit more describing for what the model is handling. Calender can be considered to be a report of CalenderEvents. See discussion of report objects in chapter 9.9 [1, p145]. This is mearly a suggestion though.

If it’s meant that the secretary is also a member (to cover the use-cases for secretary) I think it would be better to have a conceptual class for secretary as well. Then the secretary can handle the Berth as all members don’t handle berths (if handle means handling booking proposals). To represent that Member and Secretary are two different roles in the system you might want to consider using super and subclasses [1, p503] where member and secretary are two subclasses for the same superclass. It also misses the representation of Secretary handling Calendar (CalendarEvents).

If you’re using the method of looking for nouns in the use-cases you can find a couple more conceptual classes that would add to the understanding of the domain and to support the requirements. I would for example also include a conceptual class for proposals (BookingProposal) as well just to make things even clearer and have associations from BookingProposal to Berth and to Boat. In the membership use-cases membership fee is mentioned as its variable part is affected by the assigned Berth so a conceptual class representing that might be interesting as well.

Larman aslo writes about naming of associations and to use descriptive names and avoid “Has” and “Uses” [1, p152] as it’s not that descriptive. Docked-at or assigned-to or something similar just to be more descriptive?

***Do you think the model has passed the grade 2 (passing grade) criteria?***

I’m sorry but even though a domain model seldom is correct from start and you shouldn’t even aim at having a completely correct domain model from start [1, p133] I think this model lacks a bit too much information and for the wrong use of attributes I think it’s not yet compliant with grade 2 at this is at this stage. It is a good start with the classes and associations it has now but is in need of being gone through a bit more to support the use-cases in the requirements with a couple more classes and associations and an overlook of how to add attributes.

# References

1. Larman C., Applying UML and Patterns 3rd Ed, 2005, ISBN:0131489062