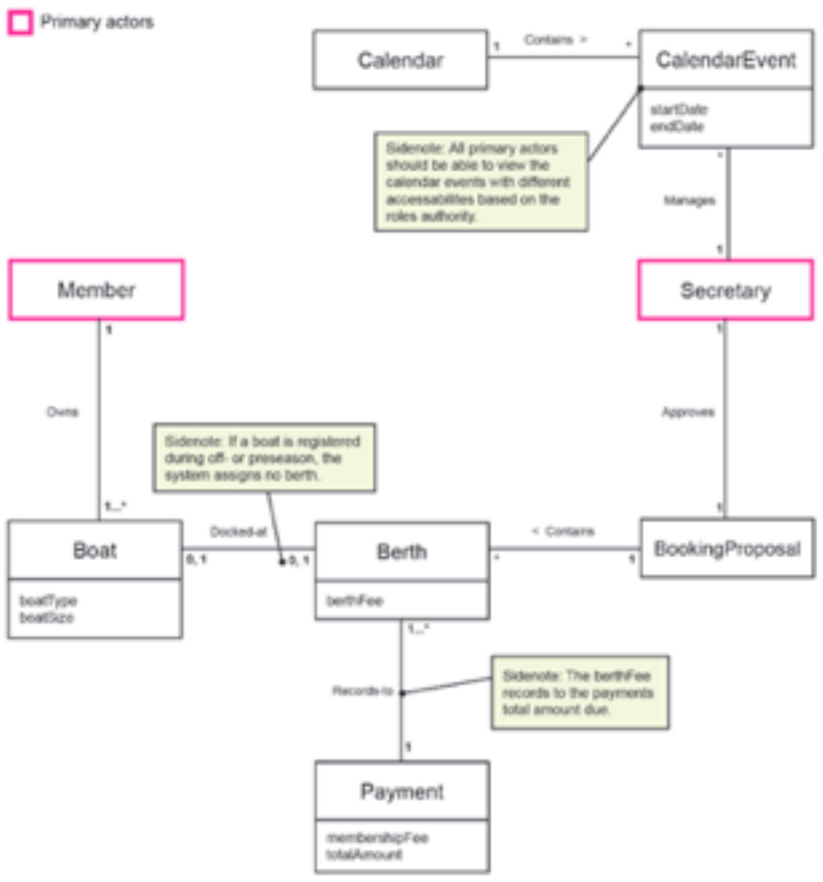
Peer review Workshop 1

For domain model made by Joakim Holmevi


Primary actors
Calendar
Contains >
CalendarEvent
startDate
endDate
Sidenote: All primary actors should be able to view the calendar events with different accessabilities based on the roles authority.
Manages
Member
Secretary
Owns
Approves
Sidenote: If a boat is registered during off- or preseason, the system assigns no berth.
Boat
boatType
boatSize
Docked-at
Berth
berthFee
< Contains
BookingProposal
Records-to
Sidenote: The berthFee records to the payments total amount due.
Payment
membershipFee
totalAmount

As a developer would the model help you and why/why not?
Yes. It would give me a lot of help understanding what's supposed to be done. It seems to have all conceptual classes needed compared to the requirements. The modeler seems to follow the guidelines to think like a map maker, mentioned by Larman, with relevant naming from the domain [1, p145]. Regarding attributes it seem to follow the guideline with selecting attributes the requirements suggests [1, p158]. It also seems to focus on the real world and not software [1, p136]. In combination with the requirements it would be a lot of help even though it may lack a couple of associations which might make it more clear (see under the question *What are the weaknesses of the model, what do you think should be changed and why?)*

Do you think a domain expert (for example the Secretary) would understand the model why/why not?
Yes. It's clear what's associated with Secretary vs Member and as the naming is good which makes it easy to follow and understand.

What are the strong points of the model, what do you think is really good and why?
The answer as is pretty much the same as for the first question. It seems to have all conceptual classes needed compared to the requirements. The naming is good which makes it easy to understand. Good selection of attributes in conjunction with the requirements. Good UML notation.

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

To show the fact that Member pays for the membership fee and berthFee(s) I think it would be a good idea to connected Payment to Member with an association (Pays-for).

A BookingProposal contains the match between Boat and Berth and I think the model should reflect that relationship with an association between BookingProposal and Boat as well.

Member and Secretary are two different roles in the system and I appreciate the thought of enhancing that with the use of pink color but I don't think it's correct UML notation. (I can't find it anywhere in the book so I assume it's not correct?) Why not use super and subclasses [1, p503]? Have Member and Secretary as subclasses and create a superclass above them as they both are representing different roles in the system.

I might have misunderstood but as I read the model I assume that totalAmount includes both membershipFee and berthFee(s). Then it's an derived attribute [1, p160]. To mark that use "/" in front of it.

I would also add the sidenotes as text outside the model instead of in the model just to follow the correct UML notation. Again I can't find any reference saying that it's wrong but I can't find it anywhere in the chapters for domain modelling so I assume it's not correct UML notation? And for the note about primary actors and calendarEvent why not have an association (Views) between Member and calendarEvent to show it in the model?

A **minor** detail is that without readingdirection-arrow between CalendarEvent and Secretary it now looks like "CalenderEvent manages Secretary" if you follow the convention [1, p152] (even though I think everyone understands that it's meant to be the opposite so I understand if this is an unnecessary comment ☺).

Calendar may be considered as an report of CalendarEvents. Larman have a discussion about report objects in chapter 9.9 [1, p145] about reasons to ant not to include report objects. It may not me necessary in the model with report objects but at the same time it might be if it's a noteworthy term. So I'm not saying it's wrong at all it might just be good to consider why to include it. ☺

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

Yes, absolutely. It has the conceptual classes and attributes needed for the use-cases. It's clear and easy to read. As a domain model seldom is totally correct for the first time [1, p133] and this model is good example for a first iteration. My reflection is that it only needs some minor corrections to be even better.

References

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