

Dear Axel Nilsson,

The current domain model is solely limited to the use cases for the grade two requirements. Therefore, the reason that you decided not to mention a treasurer in the submitted model is assumed as the treasurer does not have any role. However, offstage actor 'Municipality' has no role inside the domain model either - according to the use cases - but it is still presented in your model. Since there is no description of specific role of the municipality from the use cases, it is considered that in the model mentioned association "gets renter of" is irrelevant to the current domain. As, irrelevant information is recommended to be removed[1]. - "Domänmodellen är också en abstraktion, dvs den visar inte hela verkligheten utan **bara det som vi tycker är intressant** för det blivande programmet" - hereby I recommend to remove the municipality.

Another thing to notice is that the association member "registers" its boats already implies the member "rents" a berth for its respective boats, which therefore these associations are **duplicated information** within the provided model. In such case, it is recommended to avoid too many associations [2, p.151]. In a similar manner, if 'Allocation proposal' assigns a boat to a berth, the boat is then assigned. Thus, to remove 'Assigned-to' association of boat to berth is recommended.

Also, It is difficult to find difference between member lists the calendar event and views it since it is visualising basically same information. Therefore unification of two associations between member and calendar event is suggested. Similarly, association that member 'removes' boat and 'changes' boat could be unified as one association, due to the same argument [2, p.151]

The 'Gets-allocation-history-of' association from a conceptual class 'allocation proposal' has not specified in use cases. It is understandable that the model tried to allocate boats in close to an arbitrary member's already registered boats, but in this sense berth should also refer whether a member has paid the membership fee or not. As purpose of domain model is to have an abstraction, instead of having too much details [1], it is recommended to remove 'Gets-allocation-history-of' association between berth and 'allocation proposal.'

Regarding to 'Person' and 'authentication', I would like to remind you that we focus on reality but not on software implement at current phases. The concept of providing ID and passphrase to be authenticated is be regarded as a software concept, instead of referring to daily situation. In a casual manner, it is already assumed as a pre-condition of current model, and therefore it is recommended to drop those part out of the model.

Finally, as many classes presented in the model miss their attributes, it is recommended to add missing information [2 ch.9.16]

References

1. Leif L., Domain model, 2010-06-16,
<http://people.kth.se/~leifl/artiklar/j2ee/projektsajt/elaboration/analys.html>
2. Larman C., Applying UML and Patterns 3rd Ed, 2005, ISBN: 0131489062

Appendix. (Questionaries provided by instruction)

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

The provided model reflects the problem description in a logical way, despite there are some minor confusions due to several duplicated information. However, it followed UML naming conventions, and usage of non-vague words helped to clarify the problem. Therefore I conclude this model is helpful for a developer

Do you think a domain expert would understand the model why/why not?

A domain expert, such as a secretary, can doubt about 'municipality' as there is no reason for it to be mentioned in the model. The expert could also wonder whether 'person' class with username/password is required. Except for these minor faults, domain expert would not have difficulties understanding the model as it is quite straightforward.

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

The naming of the association is well chosen, as it follows naming rule suggested by Larman C., and therefore this is one of strong points of the model.

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

The model has some unnecessary associations, which leads to confusion and disruption. For example, secretary manages a calendar event, which may imply both creating a new event, modifying information of event, and deleting it. I wonder if there was a specific reason for associations between member and boats to be separated as 'removes' and 'changes'. Also, some association is duplicated, which could be simplified by removing them. Furthermore, many classes are missing attributes which I thought should be included as some were mentioned within use cases.

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

Yes.