

Workshop 1 - Domain Modeling

Comments on received peer review

by Daniel Vedin

The situation requirements between "Member" and "Boat" are well described in the use cases already, and won't be of any help understanding the conceptual model and the relations between the conceptual classes. The association that shows that the member actually owns the boat describes the real-world domain. Methods that the owner can do to the boat within a system describes the software domain.

About the tricky relationship between "Boat" and "Berth": I agree that it makes it tricky, but still it describes the real-world. A berth should be able to exist without a boat in it, and during off- and preseason the registered boats assigns no berth. I can agree that all the boats needs one berth anyhow, and that the off- and preseason exception is describes well enough in the use cases.

Thank you for teaching me how to write the derived attributes for the conceptual classes.

About the boatSize attribute lacking information about it's value is because the use cases doesn't say which value it should be. But I agree with you that it should show nevertheless, so I added a measurement to it.

by Monica Sundin

I agree that an association should be set between "Members" and "Payment", but the use cases didn't really showed that association for the passing grade 2. Anyway, the association is pretty obvious so I added it to make the domain model more complete.

About the missing association between "Boat" and "BookingProposal" I can agree that in a software model it should make it more clear, but in the real-world domain the boat is actually docked at the berth, and the booking proposal contains all the berth with it's associative boat. I added the attribute "boatID" to the "Berth" to make it clear that the associated boat somehow adds to the "BookingProposal".

Thank you for learning me about the derived attributes for the conceptual classes.

The association between "Member" and "Calendar" I agree to. I guess I got a little confused on the last workshop where Tobias asked if that association is necessary to have. But then I had the relation 1 to 1, and i agree that all member should be able to view the calendar.

Thank you for bringing the report objects up in the peer review. I added the "Calendar" mostly because I couldn't find any references on many to many relations between conceptual classes. That's why I needed the "Calendar" class to collect all the events that all members can read, so I didn't have to use a many to many relation between "Member" and "CalendarEvent".