Peer review Daniel Vedin

2015-09-17

Authors: Jakob Brandt, jb223df@student.lnu.se
Thorvald Gudjonsson, tg222eu@student.lnu.se
Mauro Pappaterra, mp222tb@student.lnu.se

- As a developer would the model help?

We would say that from a developers perspective this model wouldn't help us very much. Since we know what kind of system this model is supposed to cover it doesn't correspond to the system we envisioned. A lot of information gets lost and trying to fit the entire system into one domain model might be hard, we would recommend to have maybe two or three or if it's one model to put in a lot more work. We couldn't find some classes that we've identified that are important to the system in the model, one example is the treasurer of the yacht club [1].

- Do you think a domain expert would understand the model?

The actual domain model that's been made isn't hard to understand, it's just too little information in it. The actor (the member) at the bottom of the model wouldn't make any sense for a domain expert in our opinion, it doesn't add any information at all. We think that it would probably just confuse a domain expert.

- What are the strong points of the model?

We think that when quantities are specified in between the relations, for example that one member can have several boats helps the understanding of the model. In the domain model (the top part) there is a nice flow and apart from some things we will discuss in the next section it is easy to follow and understand. Also some relations are well named, the two realtions we had in mind in are “provides-information-about” and “is-assigned” [2].

- What are the weaknessess of the model?

The biggest weakness is the actor (the member) at the bottom of the model, it really just creates confusion. We perceive it as if the author is trying to make an actor UML diagram which is not part of the domain modeling. We feel since the convention is to read from left to right and up down, the relations should be clarified with an arrow when the case isn't that you are to read from left to right and make the lines between classes perpendicular [3][4].

- Do you think the model has passed grade 2?

Since the task was to model a set of requirements that we don't see modeled in the domain model and almost half of the requirements are covered with an actor diagram which isn't a part of domain modeling, we would say no.

Reference

[1] http://coursepress.lnu.se/kurs/objektorienterad-analys-och-design-med-uml/workshops-2/workshop-1-domain-modeling/

[2] “Applying UML and patterns”, by Craig Larman, (9,14 Applying UML: Multiplicity), page 153-154

[3] https://en.wikipedia.org/wiki/Actor_%28UML%29

[4] “Applying UML and patterns”, by Craig Larman, (9.14 Apply UML: Accociation), page 151-152