Workshop 1 Domain Modeling

Peer review for rk222ev

Strong points

You have really been thinking like a map-maker. Larman states on page 138: ”Use the existing names in the territory”, and makes an example with borrowers and patrons as conceptual classes for a library domain model. Likewise, it's good that you have used Berth as conceptual class, like in the requirements, instead of something else like Dock or Parking spot.

It's also good that you focus on the static information needed to support requirement, and leave out the dynamic view of the secretary that has already been handled in the use cases. This is good because, according to Larman on page 129, ”The model displays a partial view, or abstraction, and ignores uninteresting (to the modelers) details.” You also left out the authentication (password, username, user). That is good because it is part of software and not the domain, and is therefore also uninteresting for the domain.

The model implements correct UML and we find it easy to read and understand, due to the absence of uninteresting classes.

Would this model help us as developers?

Yes. It gives an overview of what is needed to support the use cases. When looking at the model, we felt that it gave clarity to how the boat club concepts were connected.

Something that also helps is to see that one member might own zero to more boats. If we were to design and implement the system, it's good that the model shows us that a member can possibly have 0 boats registered.

Weaknesses

It might be good to show the multiplicity between the Boat and Berth classes. As Larman says on page 159, ”The multiplicity value is dependent on our interest as a modeler and software developer, because it communicates a domain constraint that will be (or could be) reflected in software”. It's good to clear up if a boat can be assigned to several berths.

Is it neccessary to have the different boat types as specialized classes? Larman, page 138: ”If we do not think of some conceptual class X as a number or text in the real world, X is probably a conceptual class, not an attribute.” Couldn't the boat type be considered as a text string in this case? And maybe wait with the generalization until the software design.

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

Yes. It's a good overview of the domain. The UML notation is correct. It passes the grade without problem.

References

Larman C., Applying UML and Patterns 2nd Ed, 2002, ASIN: B001BARLXM
http://www.utdallas.edu/~chung/SP/applying-uml-and-patterns.pdf