Workshop 1

Peer Review of Workshop 1 for Daniel Blixt and Daniel Nilsson

Looking at the domain model made by aforementioned group we can see that their design is very specific and detailed, often using patterns you would see in a database model, such as breaking out attributes into their own concepts to avoid duplicating data. We can also see many concepts that we did not immediately recognize as part of the problem description, we'll get into why we disagree with some of them later.

Critique:

The analysis part of the diagram should mainly include the most basic concepts that are described in the problem description [1]. Craig Larman has an example of concepts such as "Library", "Book" and "Patron" in a library information system. These are very simplified concepts for an obviously much more complicated situation and we feel that the diagram supplied to us here was very complicated compared to the examples set by C. Larman.

They also have associations that specify exactly what kind of interaction is done between the concepts which isn't part of the domain model but rather part of an interaction diagram. The domain model should instead just specify the most noteworthy relation between concepts, such as a "Boat" being 'owned' by a "Member". The diagram supplied to us was much more software oriented which is something that isn't needed before the design part kicks in - where software objects are identified from the concepts from the analysis.[2]

Things to be improved could be removing concepts such as BoatSize, CalenderEvent or BerthDescriptor as these could just belong to the main concept Boat, Calendar and Berth. The concept BoatClub could be removed as it doesn't add anything to support the use cases for grade 2. We don't need it to be explicitly explained that the member belongs to a BoatClub, this is implied and this is more of a software implementation point of view. Same goes for the authentication part of User and UserRegister.

While we personally would prefer a more detailed model (as it seems most students have) you sadly cannot argue with C. Larman and Tobias Ohlsson, they run a tight ship.

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

As a developer this model would be very helpful as it provides good information for building a structure in an object-oriented language. You could start up a project and implement many classes as they're listed in the model.

- 1. Larman C., Applying UML and Patterns 3rd Ed, 2005, ISBN: 0131489062, (1.4)
- 2. Larman C., Applying UML and Patterns 3rd Ed, 2005, ISBN: 0131489062, (1.5)

Do you think a domain expert (for example the Secretary) would understand the model why/why not?

Most likely not since she's not a developer and would not understand the point-of-view of a software developer. She may have some understanding of it but most likely many parts of it would be unnecessarily confusing for a non-developer.

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

The concept and association names are mostly very clear and as mentioned above, you could implement this into code with relative ease as the concepts/classes are already staked out well. It manages to be readable although it is quite complicated.

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

Not understood easily for non-developers, too complicated and detailed for a domain model. BerthDescriptor is its own concept although its attributes could have been displayed in Berth (making BerthDescriptor unnecessary). Same goes for Boat and Calendar.

We think that this model was too influenced by software and database modeling and some of the concepts are entirely software-based (such as User/UserRegister) and these should be removed for the benefit of a simpler domain model.

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

We think it contains the correct material for passing grade 2, but it has to be changed slightly to remove the software artifacts and concepts and simplified to fit in the pattern of a domain model.

Adam Österlund, Kevin Amilund, Jonas Tornfors.

- 1. Larman C., Applying UML and Patterns 3rd Ed, 2005, ISBN: 0131489062, (1.4)
- 2. Larman C., Applying UML and Patterns 3rd Ed, 2005, ISBN: 0131489062, (1.5)