Åsa Axelsson
2016-09-15

Peer Review, Workshop 1

Group: Anton Månsson (am222xb), Mikael Andersson (ma223im)
Link: https://github.com/am222xb/1dv607-workshop_1

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

At first sight, without looking at the content, the model gives a good impression. You can easily see the starting point and follow the boxes downwards.

"Secretary", "Treasurer", and "Member" are displayed in a horizontal line, which gives a good understanding of the different user roles. And it is easy to follow their different tasks.

The overall understanding of the domain model is good but as a developer you probably will question the lack of association names and multiplicity expressions. The UML notation is not standard and can make the user confused.

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

The Secretary would probably understand the model. But from a "map makers" perspective the model could be even more simplified. Now the model includes some classes that are more software-related than real-world-situations.

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

- The overall visual appearance is good.

- The model does not include any attributes and from a "map maker" perspective I really like this idea. This makes the model really easy to understand for a non-expert. However, from a developer's point of view this could probably be debated.

  [1] Larman, chapter 9.16, p158:

  "Include attributes that the requirements suggest or imply a need to remember information."

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

Association naming

- The association naming should be “Manages” instead of “Manage berths “.

  [1] Larman , chapter 9.14, p 152
  “Name an association based on a ClassName–VerbPhrase-ClassName format where the verb phrase creates a sequence that is readable and meaningful.”

- One association text is displayed vertically. I believe I would have noticed this text better if it was written horizontally.

Class naming

- Use singular for the class Berth.

Multiplicity expressions

- Multiplicity expressions are according to Larman optional to use. In your model you only show a single multiplicity value next to the Boat class. From a developer perspective I think it is good to show all multiplicity values in the model. Now the developer might ask: Can a Boat have many Berths?

  [1] Larman, chapter 9.14, p154:

  “The multiplicity value is dependent on our interest as a modeler and software developer.”

Do not include to much information

- The classes “Sign up”, “Log in information”, and “Database” are more software related. How essential are these parts for the understanding of the domain?

  [1] Larman, chapter 9.18, p168:

  “A useful domain model captures the essential abstractions and information required to understand the domain in the context of the current requirements.”

  [1] Larman, chapter 9.2, p134:

  “The term “Domain Model” means a representation of real-situation conceptual classes, not of software objects.”

Do not add things that doesn't exist

- "Treasurer" could be excluded as the role is not part of the requirements for grade 2.

  [1] Larman, chapter 9.10, p145:

  "Exclude irrelevant or out-of-scope features.."

Correct UML notation

- The model does not follow the correct UML notation for associations. The lack of the association names make the model unclear.

- Using arrows in both ends, for example between "Sign up" and "Login information", is confusing. What does it mean?

  [1] Larman, chapter 9.14, p151:

  "An association is represented as a line between classes with a capitalized association name."

Requirements missing

- In the requirements a Membership fee is mentioned. This is something that might be useful in the model.

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

All requirements for grade 2 are covered in the model. The model is kept simple and the overall understanding is good. Correct the UML notation and you will have a passing grade. Good job!

Reference

[1] Larman C., Applying UML and Patterns 3rd Ed, 2005, ISBN: 0131489062