Robin Reijo (Workshop 1)

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

A) Overall, model is not helpful to implement because there are some major problems in it. For instance Conceptual class and attribute are mixed (Event is added as attribute rather than conceptual class), yet more critically it has poor and irrelevant associations (Member 'own' Boat, Member 'has' Calendar, Secretary 'update' Berth), and 'irrelevant and duplicate attributes (Inside Secretary class: 'amount of berth' and 'cost'), which makes it difficult to understand the model In addition, association between Boat and Secretary is missing.

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

B) It will be difficult for the Secretary to understand the presented model. For instance, there is no association which describes the relationship between Secretary and Boat. In addition, model only shows that Secretary can only updates the Berth ,but according to the requirements Secretary can also assign the Berths. At last, model is describing that Secretary need to remember 'amount of berth' and 'cost' which needs to be reconsidered because Secretary as an actor needs to remember only 'username' and 'password' to login and, it is theSystem which will remember the 'assignment rules' for Secretary.

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

A. The strong point of model is that it contains almost all the important classes which are needed to fulfil the requirements for grade 2.

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

A) Following are the weakness in the model including their solutions:

1. <u>Interface (Authentication)</u>

- It is not applicable to add interface in the Domain model. According to Larman C. "A domain model is a visual representation of conceptual classes or real-situation objects in a domain. [1, p223-224]"

2. Associations

- Associations are missing. For example, a connection between ‘Secretary’ and ‘Boat’ does not have any association. Similarly, between ‘Authentication’ and ‘Member’. As we know that ‘Secretary’ plays an important role in this model, so by not showing association, the model is creating confusion about the relation between ‘Boat’ and ‘Secretary’. However, it can be avoided in some cases where the relationship does not need to be remembered or less important [1, p246-247]. In the given case, the relation between ‘Authentication’ and ‘Member’ could be avoided.
- Some associations names are not suitable. For example, the association ‘Has’ is a poor choice [1, p248-249]. Also, according to requirements ‘Member’ can ‘view Calendar’ not ‘has Calendar’.
- The other association which is used as ‘update’ is not fully explaining the role of the ‘Secretary’. Furthermore, the name must be based on the ‘ClassName-VerbPhrase-ClassName’ format [1]. Therefore, ‘assign’ is a better choice than ‘update’. Similarly, ‘occupies’ can be changed to ‘moored’.
- The word ‘own’, as an association for describing the relation between ‘Member’ and ‘Boat’, is not fully explaining the relation which is demanded in the model. This does not show that ‘Member’ can register’ a ‘Boat’ as well as update its details. It’s not necessary to show the ownership in this case because the main requirement is the ‘registration process’. So, in order to make it more abstract and understandable, ‘Multiple Association’ between two classes such as ‘Register’ and ‘Manage’ could be used[1, p251]. Therefore, we recommend to remove the association between ‘Secretary’ and ‘Boat’ as a ‘Member’ registers the ‘Boat’ and ‘Secretary’ only assigns the ‘Berth’ for the ’Boat’ according to the workshop description.

3. Conceptual classes Vs Attributes

- ‘Event’ is found to be used as an attribute in the ‘Calendar’ class. However, the workshop’s user case 11 and 12 describes that the “Calendar event is presented with a short title, start date and end dates”. So ‘Event’ itself suits better as a class because it have all the above attributes in it [1, p241]. This issue could be resolved by creating two classes named as ‘Calendar’ and ‘Event’ and join them with an association called ‘Contains’. However, from a map maker point of view you can skip the ‘Calendar’ class because ‘Events’ are in the ‘Calendar.’

4. Attributes

- There are some attributes which does not fit or unnecessary in the classes. For example, 'amount of berths' and 'cost' in the 'Secretary' class is creating confusion, because 'Secretary' does not need to remember these costs and other details, because System(user case 8) remembers all these for 'Secretary'. Therefore it is recommend to remove these attributes.
- Instead, as 'Secretary' need to authenticated so 'username and password' according to the requirements, these can be added as its attributes [1, p256], as a 'Member' class has. Similarly, it can be questioned whether 'Berth' class would need 'cost' attribute. Instead, attributes such as 'size' and 'location' could be more suitable. Also, in 'Member' class an attribute which describes 'member's data' is missing which eventually needs to be updated in use case 4, 5 and 6. Thus, I would recommend to add the important requirements as an attribute [1, p256].

5. Associations - dependency

- Some associations need dependencies to satisfy the information requirement [1, p246]. Corresponding to above statement, it is recommended to add dependency when 'Secretary' assign the 'Berth' because during registration of boat (user case 4), the assignment of the berths is dependent on the 'current assignment rules (including the current season)'. This could be done by creating a new class and join it with the association by dotted lines (represents dependency).

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

A) No, because
1. Model has poor and irrelevant associations which makes it not comprehensive.
2. Model does not fulfil grade 2 requirements, such as 'register a boat'. Also, the model does not clearly explains some other requirements such as 'View/List Calendar Event (by member)' because model describes that Member 'has' Calendar', which is difficult to understand the intension of doing so.

References

[1] Larman C. , Applying UML and Patterns 3rd Ed, 2004, ISBN: 0131489062