Workshop 1 - Domain Modeling

Peer Review of Frida Holmström (fh222dt)

First of all I want to make it clear that I took the references from a PDF version of the coarse literature because I have the physical book stored in a box on a farm somewhere. That means that the page numbers might not be the same as in the physical book, unfortunately. The chapter numbers should still be the same.

As a developer would the model help you?

I'm afraid that the model wont help me as a developer. I feel that it's too much missing information from the use cases and requirements that still needs to make the domain model clear for me. The object-oriented analyses should be concerned with creating a description of the domain from the perspective of objects with an identification of the concepts, attributes and associations that considered noteworthy (Larman, 2004, p 43, chapter 1.5). An useful technique to get the noteworthy conceptual classes from the use cases that Larman (p 232, chapter 9.5) describes is by doing a linguistic analysis. Identify the nouns and noun phrases in the use cases of the domain, and consider them as candidate conceptual classes or attributes. Larman (p 226, chapter 9.2) also says that a domain model is not a data model, so you shouldn't exclude a class simply because the requirements don't indicate any obvious need to remember information about it of because the conceptual class has no attributes.

Do you think a domain expert would understand the model?

Not really. I think the expert first would say that the domain model is a false visualization or the real-world domain (Larman, 2004, p 44, chapter 1.5). For example, the Members should not be able to add, remove or change an event i the Calendar, but a secretary does. What exactly does the Member handles in the Berth? That association is not very clear. Larman (p 249, chapter 9.14) also says that simple association names such as ”Has” or ”Uses” are usually poor, as they seldom enhance our visualization. Association names should also start with a capital letter Larman says, since an association represents a classifier of links between instances; and in UML, classifiers should start with a capital letter.

What are the strong points of the model?

The strongest point of the model is that it's clean and it uses good names for the conceptual classes.

What are the weaknesses of the model?

The biggest weakness is that in contrast to the domain model showing the real-world classes, this diagram shows software classes (Larman, 2004, p46, chapter 1.5). The diagram should be interpreted as describing things in a situation of the real world or domain of interest. Check the figure 1.6 (Larman, 2004, p49, chapter 1.6) to see the difference between a domain model (conceptual perspective) and a design class diagram. Larman (p 223, chapter 9.2) says that the term Domain Model means a representation of real-situation conceptual classes, not of software objects. That means that the responsibility methods that are used frequently in this domain model, should not be shown until the design model. The conceptual model (domain model) should only show:

- Domain objects or conceptual classes
- Associations between conceptual classes
- Attributes of conceptual classes

Do you think the model has passed the grade 2 criteria?

That's not my job to decide.

References

Larman, Craig. (2004). *Applying UML and Patterns: An introduction to Object-Oriented Analysis and Design and Iterative Development, Third Edition*. Addison Wesley Professional