Problem description

> On the last lecture, I have argued with the lecturer about whether sequence diagram should be in the Logical view or Process view in 4+1 view model of the system.
> That was a long argue but in the end, Dr. Truong Tuan Anh conclude that: any diagram can represent any view, but sequence diagram should be more in the side of logical view, instead of process view in reference to the ref book and reality application. Also in the process view, the activity diagram is superior to sequence diagram when representing Process view.
> I was quite confused and unsatisfied with the answer so I dig more into the reference book for more information.

View of Iam Sommerville

First, let's have a look at our reference book of Iam Sommerville. On page 175 he wrote:

> There are differing views about whether or not software architects should use the UML for describing and documenting software architectures. A survey in 2006 (Lange, Chaudron, and Muskens 2006) showed that, when the UML was used, it was mostly applied in an informal way. The authors of that paper argued that this was a bad thing.
>
> I disagree with this view. The UML was designed for describing object-oriented systems, and, at the architectural design stage, you often want to describe systems at a higher level of abstraction. Object classes are too close to the implementation to be useful for architectural description. I don’t find the UML to be useful during the design process itself and prefer informal notations that are quicker to write and that can be easily drawn on a whiteboard. The UML is of most value when you are documenting an architecture in detail or using **model-driven developmen**t, as discussed in Chapter 5

- As he said UML is for OOP system & Object, classes are too clost to the implementation → not useful for architectural description.
- Model-driven development is a type of System modelling that "is a model-focused approach to software design and implementation that uses a subset of UML models to describe a system" said in *Chapter 5.5: System modeling-Model-driven architecture* . The architecture said in here is just a name but not the architecture of the software that we are talking-studying about.

- Model-driven engineering is an approach to software development in which a system is represented as a set of models that can be automatically transformed to executable code

→ We now know that the author of the reference book, is not found UML useful during the design process of the architecture and "prefer informal notations that are quicker to write and easily drawn on a white board"

Conclusion

In the book, we have no clue of which UML diagram is used for which view in the process of software's architectural design. Furthermore, the author does not suggest using UML for software architecture design.

But in the reference part of the book, we could find some useful information about this discussion.

UML: Software Engineering with Objects and Components

- In *Chapter 14: Packages & models*, section *14.2: Models* the author argued:

  > The UML models don’t map exactly onto the 4 + 1 views, but they do cover the necessary aspects. To recapitulate:
  >
  > > **The logical view** is taken by the class model, and by interaction diagrams and state diagrams to the extent that they are used to specify the logical behavior of the system;
  > >
  > > **The process view** is taken by interaction diagrams, state, and activity diagrams, to the extent that they are used to determine the threads of control of the system, and by deployment diagrams;
  > > *Page 158*

- Now, we know that, when using UML for 4+1 view of a system, we should consider
  - **Logical View** as
    - Class model (diagram)

  - interaction diagrams and state diagrams: in case we use that to specify the logical behavior of the system
- **Process View** as
  - Interaction diagrams, state, activity diagrams to the extent that they are used to determine the threads of control of the system
  - deployment diagrams

> **Definition: **interaction diagram** in the book**
>
> UML provides two main sorts of **interaction diagram**, **sequence and communication diagrams**
> *page 112*

Other sources on the internet

Sung, Wook Ahn, Yong Lee Nam, and Yul Rhew Sung. "A Conceptual Model of Knowledge Management System by using" 4+ 1" views of UML."

https://koreascience.kr/article/JAKO200011922690994.pdf

- **Section 4.3 Process view**, page 130:
  - A sequence diagram shows how objects interact with each other. It takes into account some nonfunctional requirements, such as performance, system availability, and system fault tolerance at the system integrators level. And sequence diagrams are for expressing process flows between classes.

Course DIT184 University of Gothenburg - Chalmes University of technology

> https://youtu.be/r8ucofil8vY

Sequence diagram and activity diagram are listed in the process- behaviour view but logical view.

Conclusion

Through all the sources that I have listed above.

- I agree with you about a diagram can represent multiple views, because
    - They are actually not tightly mapped and UML is actually not built for architecture design, based on both books that I mentioned.
    - But we can still use the UML system because it is more readable and understandable.
- But I disagree with you about later the idea because
    - In the reference book on which you confirm the slide is based, it is obvious do not mention anything about the relationship between UML diagram and 4+1 view model. Furthermore, he disagrees with using UML for architectural design.
    - In the book "UML: Software Engineering with Objects and Components",to which the book "Software engineering" of Sommerville references.
        - The authors list Sequence diagram (as interaction diagram) beside activity diagram in Process view taking that " they are used to determine the threads of control of the system".
        - → Although Sequence diagram can also be used in Logical view, I don't think Sequence diagram is inferior to Activity diagram when representing Process view.
        - In term of reality application, I also don't think Sequence diagram is any inferior to Activity diagram due to the article that I attached. They are actually using sequence diagram to represent process view.

Thank you for reading all of this. If any of my idea is wrong, I hope you can explain it further in the next lecture. If not, I hope you can fix the slide and explain for everyone about this confusion.

Thank you again.