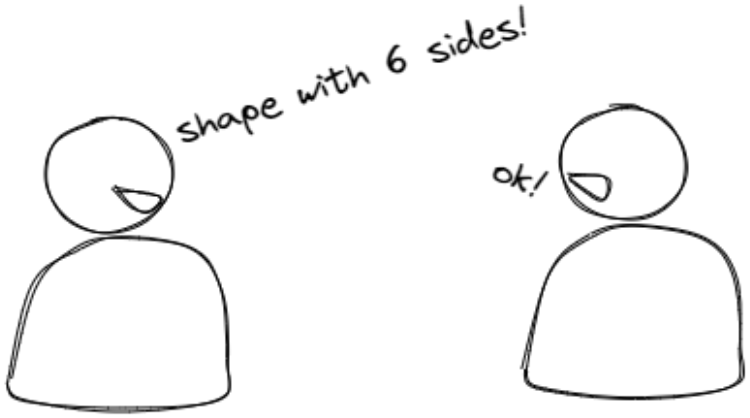
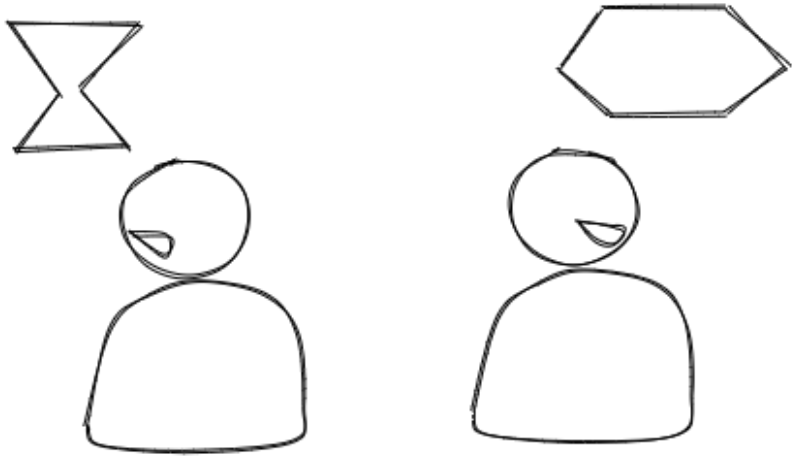
Sequence diagram

Introduction to UML

- Our work includes a lot of talking
- Talking to stakeholders or peers can be quite hard!

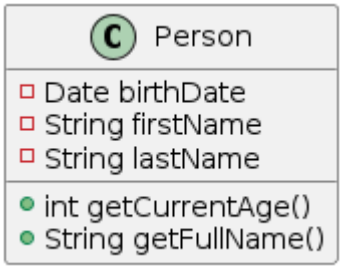




Unified Modelling Language

- Standardized method for creating diagrams
- Has 14 official diagram types
- Due to Babylonean practices, we have dialects!
- Still great tool for aligning ideas

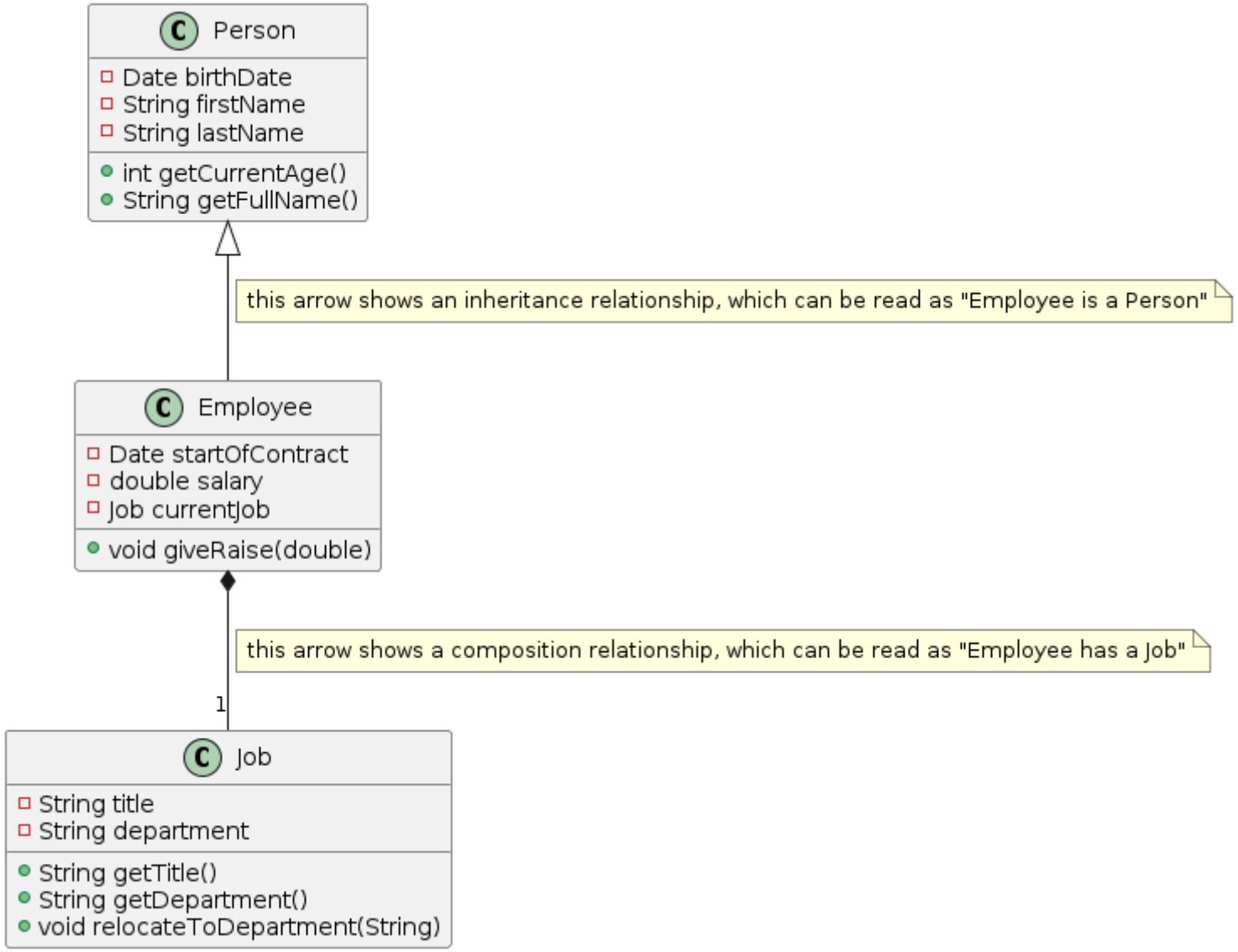
The Class Diagram



- Anatomy of a Class
    - Classname
    - Fields
    - Methods

NOTE	Fields and Methods are collectively called *members*

The Class Diagram : 2

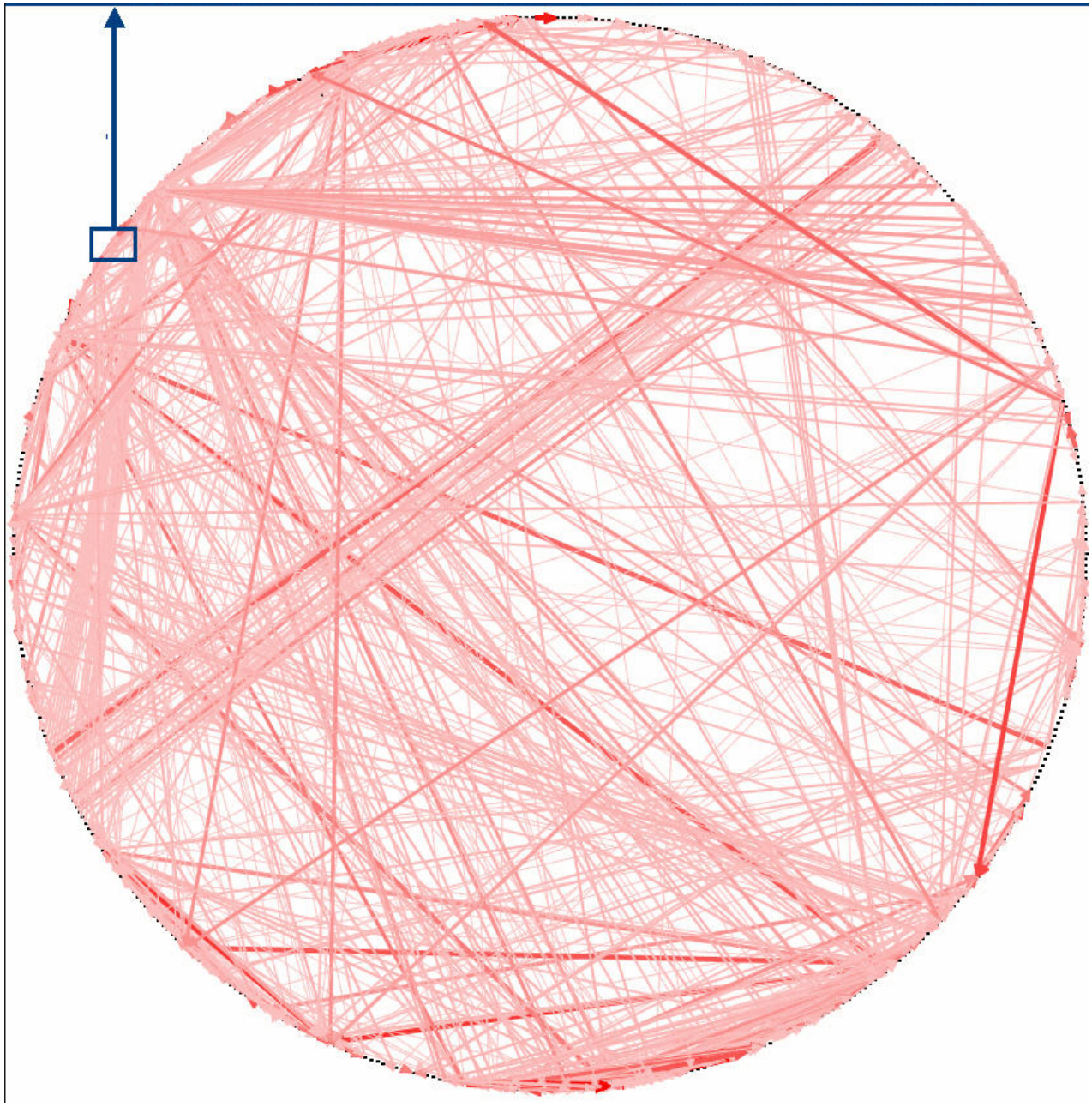


- Multiple classes show relationships

- Different types of relationships have different arrows
- Two most-used relationships are …
    - Inheritance - `Employee` *is a* `Person`
        - `Employee` inherits all members from `Person`
    - Composition - `Employee` *has a* `Job`
        - `Employee` has a field of type `Job`, referencing a `Job`

More Classes!

- When too many classes are in one diagram …



Exercise

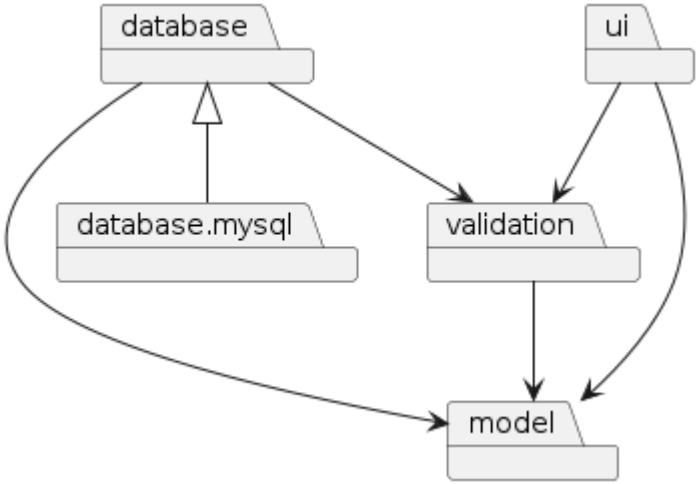
- Try creating a Class Diagram from the classes you identified in the previous chapter
- Feel free to use tools, like …
    - Draw.io
    - Planttext.com

Grouping Classes

- Grouping packages by functionality
- Grouping packages by "closeness"
    - This can also be called *cohesion*
- There is no single right way to group classes

WARNING	If someone reports a bug in your class two months from now, you will have to find that class again. Which package makes finding it easier?

Package Diagram



- Packages represent a group of classes
- Packages can *use* other classes (regular arrow)
- Packages can *extend* the functionality of other packages

Summarizing UML

- UML is great for aligning ideas
- May more diagrams
    - Structural Diagrams
    - Behavioral Diagrams

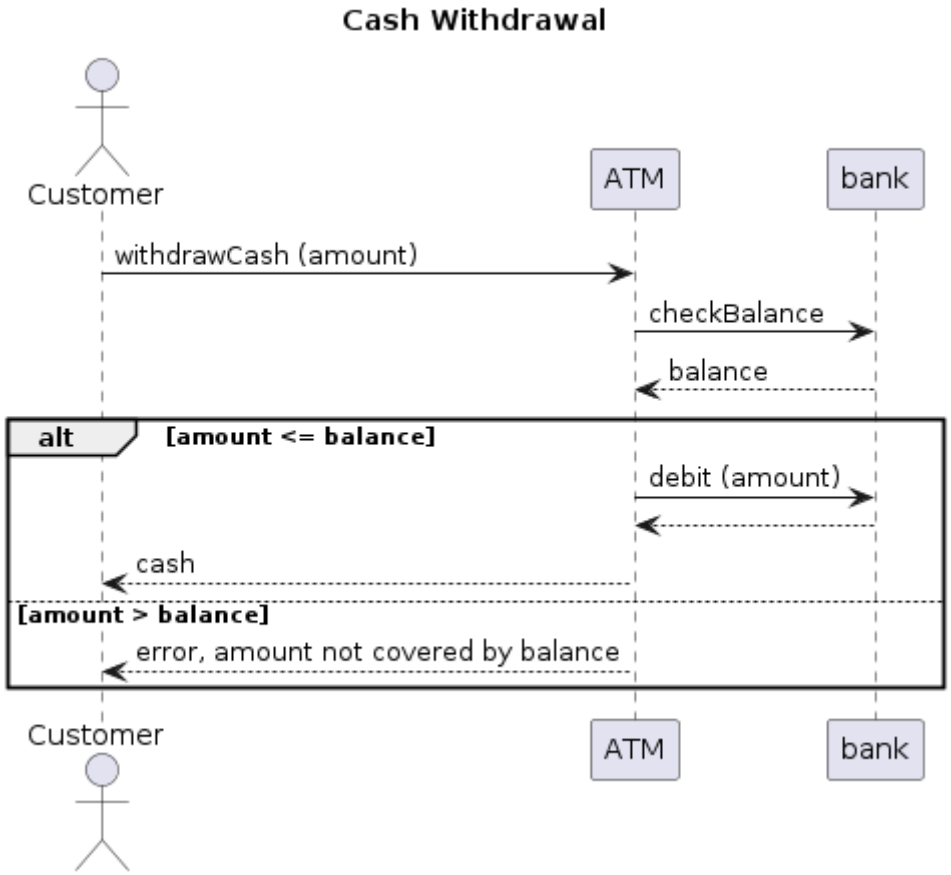
Intro

Sequence diagrams - one of many types of UML diagrams.

Sequence diagrams describe (possible) interaction between different actors ordered in time.

Here an example:

- Cash withdrawal



[1]

Reasoning and further examples

Sequence diagram can serve well, when trying to understand complex interactions between actors (objects, (micro-)services, departments) or tracing code execution as in examples below

- Desing-first (diagram serves as input for design and code)

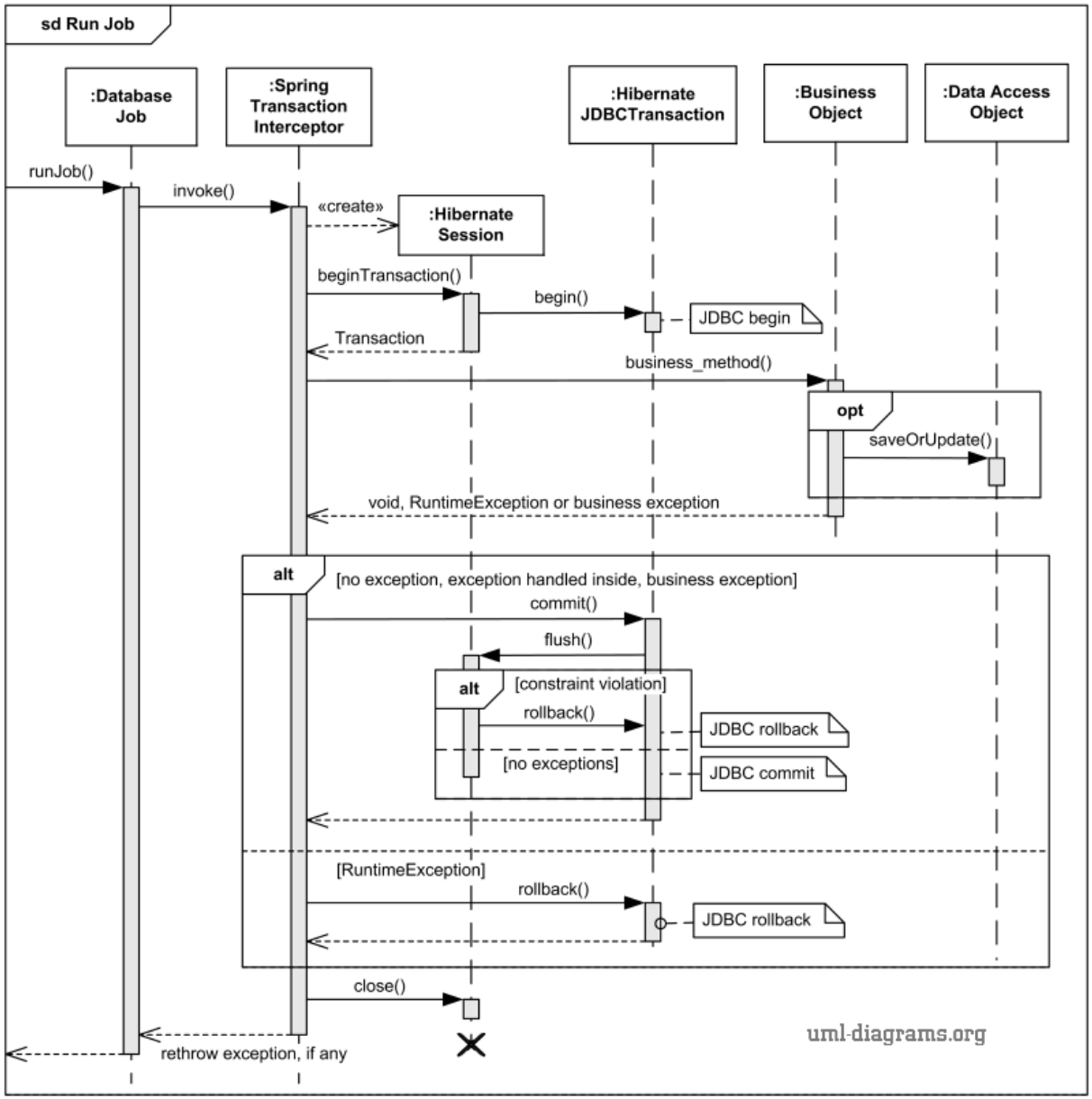


Figure 1. Spring and Hibernate Transaction

[1]

- Code-first (diagram is reverse-engineered to help understand the code flow or execution time under certain conditions)

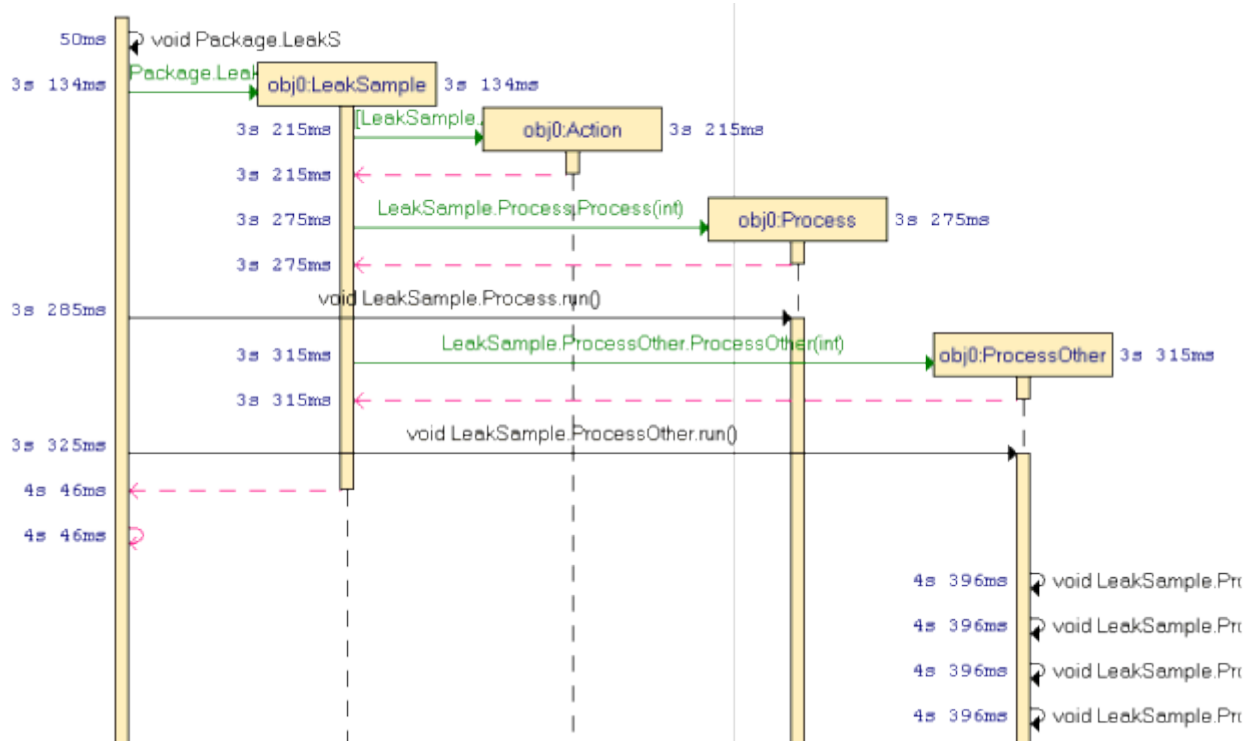


Figure 2. Runtime Tracing

[1]

Spec

Sequence diagram generally consist of

- actors (headers of lifelines ~ parallel vertical lines)
- messages (methods) (horizontal lines with arrows)
    - evtl. with parameters
    - evtl. typed
- return values
- notes/remarks

There are certain specific messages, like

- create
- destroy

which control the lifeline (vertical lines), and further extensions (UML can be very extensive and verbose), but we will leave them out of scope for now.

Fragments

UML2 allows for grouping the interactions into fragements.[2]

From the previous examples you already know

- `alt` - conditional fragments, allowing for selecting proper/desired interaction based on previous return values
- `loop` - loop fragments, allowing for repeating interactions defined amount of times or while/until a certain condition is satisfied

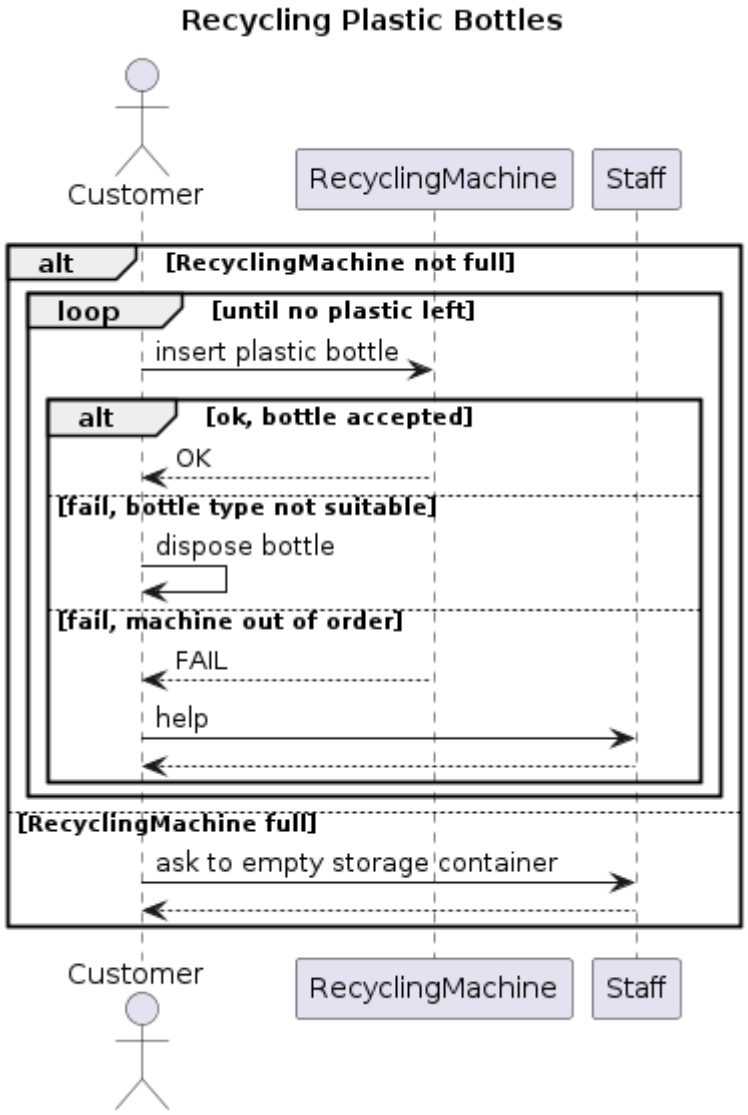
For more information please check the UML specification (section 14.3.15) or PlantUML examples under https://plantuml.com/sequence-diagram#425ba4350c02142c

---

Exercise

- Recycling plastic bottles

```
Create a sketch of sequence diagram representing interactions of customer and a plastic bottle
recycling/collecting machine. Remember to call the staff in case of malfunctions.
```



- Single Sign-On

```
Transform given activity diagram into sequence diagram
```

[1]

Further reading

- https://en.wikipedia.org/wiki/Sequence_diagram
- https://en.wikipedia.org/wiki/Activity_diagram
- https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/
- https://www.smartdraw.com/sequence-diagram/

[1] https://creately.com/blog/diagrams/sequence-diagram-tutorial/

[2] https://www.omg.org/spec/UML/2.4.1/Superstructure/PDF