Timing Diagrams

Topic # 14

Timing Diagrams

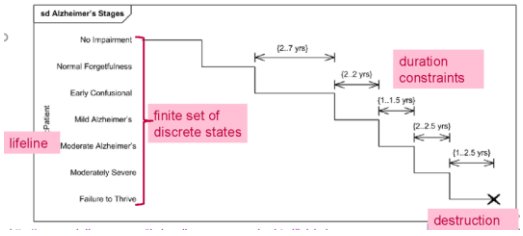
- Timing diagram is another view on the interaction model.
- A timing diagram allows you to show the interaction of objects and changes in state for those objects along a time axis.
- A timing diagram provides a convenient way to show active objects and their state changes during their interactions with other active objects and system resources.
- The X-axis of the timing diagram has the time units, while the Y-axis shows the objects and their states.
- Timing diagrams describe behavior of both individual **classifiers** & **interactions of classifiers**, focusing attention on time of events causing changes in the modeled conditions of the lifelines.

Elements of Timing Diagram

- The following elements are drawn/shown in a timing diagram:
  - Lifeline
  - State or condition
  - Duration constraint
  - Time constraint
  - Destruction event
  - Message
  - General value lifeline

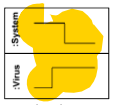
Timing Diagram Elements

- Timing Diagrams are Interaction diagram for reasoning about time.
- **Basic elements**: lifelines, states, duration/time constraints, destruction, events, messages



Lifeline

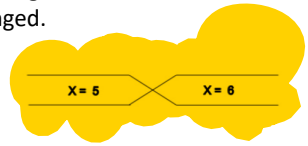
- **Lifeline** is a **named element** which represents an **individual participant** in the interaction.
- Lifelines represent **only one** interacting entity.
- Lifeline on the timing diagrams is represented by the **name** of classifier or the instance it represents.
- It could be placed inside diagram frame or a "swimlane".



representing instances of System and Virus

General value lifeline

- Shows the value of the connectable element as a function of time.
- Value is explicitly denoted as text.
- Crossing reflects the event where the clue is changed.

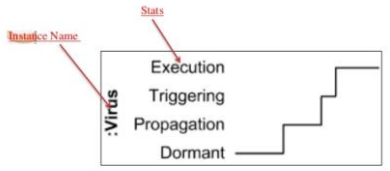


State or Condition Timeline

- Timing diagram could show **states** of the participating **classifier** or attribute, or some testable **conditions**, such as a discrete or enumerable value of an attribute.

- *Timeline shows Virus changing its state between Dormant, Propagation, Triggering and Execution state*

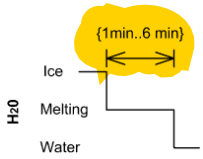
State or Condition Timeline



Timeline shows Virus changing its state between Dormant, Propagation, Triggering and Execution state

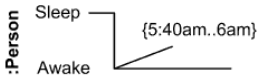
Duration Constraint

- **Duration constraint** is an **interval constraint** that refers to a **duration interval**.
- The duration interval is duration used to determine whether the constraint is satisfied.
- E.g., *Ice should melt into water in 1 to 6 minutes*



Time Constraint

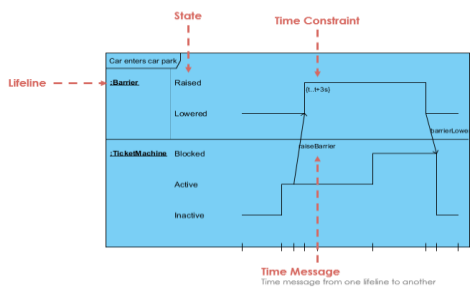
- **Time constraint** is an **interval constraint** that refers to a **time interval**. The time interval is time expression used to determine whether the constraint is satisfied.
- Typically this graphical association is a small line, e.g., between an occurrence specification and a time interval.
- E.g. *Person should wake up between 5:40 am and 6 am*



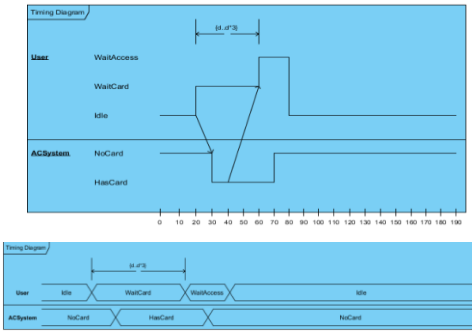
Destruction Occurrence

- **Destruction occurrence** is a **message occurrence** which represents the destruction of the instance described by the **lifeline**.
- It may result in the subsequent destruction of other objects that this object owns by **composition**.
- No other occurrence may appear after the destruction event on a given lifeline.
- ***Notation***
- The destruction event is depicted by a cross in the form of an **X** at the end of a timeline.

Carpark Example



Another Example

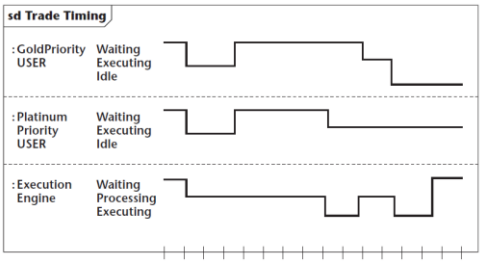


Example 1

- Following figure shows a case where two active objects share a common resource. In this case, both objects show a trade that requires the execution engine for some time. One object is for a “platinum” user who is guaranteed a trade within 10 time units, and the other is a “gold” user who has no performance guarantee. It might might wait within execution. The user objects have a waiting and executing state along with an idle state. The execution engine has a processing state that takes five time units for each trade and an executing state that takes two time units.

Example

- Different priority objects shown on a timing diagram.

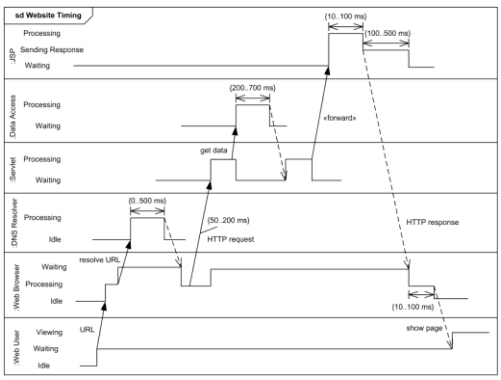


Example : Website Latency

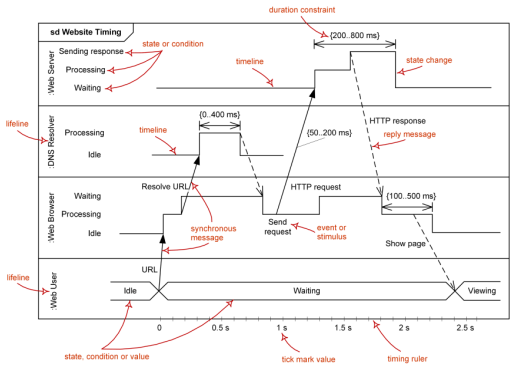
- An example of **timing diagram** which shows some **duration constraints** for a fabricated website to evaluate how long web user should wait to see something rendered on his/her display.
- After web user enters web page URL, the URL should be resolved to some IP address. DNS resolution can add some tangible waiting time to the response latency as perceived by user. Latency delays related to DNS resolution could range from 1 ms (local DNS cache) to several seconds.

Example: Website Latency

- With simple Model–View–Control (MVC) implementation, Java servlet gets control and requests some data from "model". Communication with data sources usually takes some discernible time. After data is received and processed, servlet forwards request processing to JSP ("view"). Buffered HTTP response is sent back to the browser.
- Web browser takes some time to process HTTP response and HTML page to start rendering the page view to the web client. (Note, that after that it could take even more time for the web browser to request other resources like CSS, JavaScript, images, which is not shown on the diagram.)



Example (website) : Notations



END OF TOPIC 14

-COMING UP!!!!!!

-Implementation Diagrams
-RUP
-4+1 Model view