Diagramming

Matthias Margush

July 20, 2019

Contents

1 Configure **2**

2 Embedding a Diagram **3**
2.1 Example 3

3 Use Case Diagrams **4**

4 Sequence Diagrams **4**
4.1 Arrows 4
4.1.1 Example Sequence Diagram Arrows 7
4.2 Participants 8
4.2.1 Example Sequence Diagram 8
4.2.2 Sequence Numbering 8
4.2.3 Page Titles, Headers, Footers 9
4.3 Sequence Grouping Boxes 9
4.3.1 Example Sequence Diagram - Alternates 10
4.3.2 Example Sequence Diagram - Loops 11
4.3.3 Example Sequence Diagram - Parallel 12
4.4 Notes 12
4.4.1 Notes relative to message 12
4.4.2 Notes relative to participant 12
4.4.3 Example Sequence Diagram With Notes 13
4.5 Formatting 13
4.5.1 Example Sequence Diagram With Formatting 14
4.6 Splits 14
4.6.1 Example Sequence Diagram With Split 15
4.7 Delays & Spacing 15
4.7.1 Example Sequence Diagram With Delays And Spacing 16

4.8 Lifelines 16
4.8.1 Example Sequence Diagram With Lifelines 17
4.8.2 Example Sequence Diagram With Auto-lifelines 18
4.8.3 Example Sequence with Shortcut Lifelines 19
4.9 Participant Creation 19
4.9.1 Example Sequence Diagram With Participant Creation 20
4.10 System Boundaries 20
4.10.1 Example Sequence Diagram With System Boundary . 20

5 Entity Diagrams **21**
5.1 Entities 21
5.2 Connectors 21
5.3 Example Entity Diagram 22
5.4 Lines 22
5.5 Arrows 22
5.6 Vertical vs Horizontal Layout 23
5.7 Relations 23
5.8 Association Class 23
5.9 Fields & Methods 23
5.10 Visibility 23
5.11 Abstract, Static, Stereotypes, annotation, enum 23
5.12 Separators 24
5.13 Notes 24
5.14 Example Class Diagram 25

1 Configure

- System: install plantuml (e.g. `brew install plantuml`)
- Emacs: install plantuml-mode. Configuration:

```
(setq plantuml-jar-path "/usr/local/Cellar/plantuml/1.2019.6/libexec/plantuml.jar")
(setq plantuml-jar-args '("-charset" "UTF-8" "-config" "~/plantuml.txt"))
(setq org-startup-with-inline-images t)
(add-to-list 'org-src-lang-modes '("plantuml" . plantuml))
(org-babel-do-load-languages
 'org-babel-load-languages
 '((plantuml . t)))
```

- `C-c C-c` in a source block to render a diagram.
- May need to `org-toggle-inline-images` to view inline.

2 Embedding a Diagram

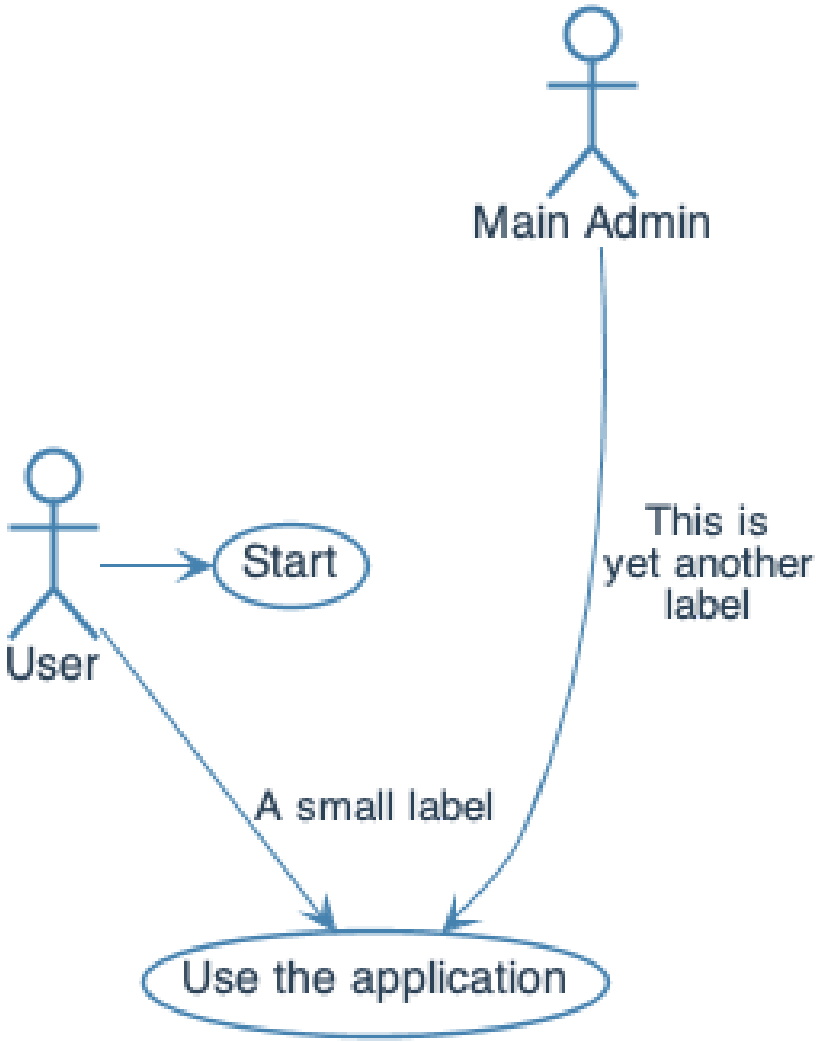
`#+begin_src plantuml :file output.png` Begin diagram

`#+end_src` End diagram

`#+STARTUP: inlineimages` Display diagrams inline (put at top of file)

2.1 Example

3 Use Case Diagrams



4 Sequence Diagrams

4.1 Arrows

-> **or** <- Solid arrow

--> **or** <-- Dotted arrow

`-\` **or** `-/` Half arrowhead

`-\\`, `-»`, `«-`, `-//` Skinny arrowhead

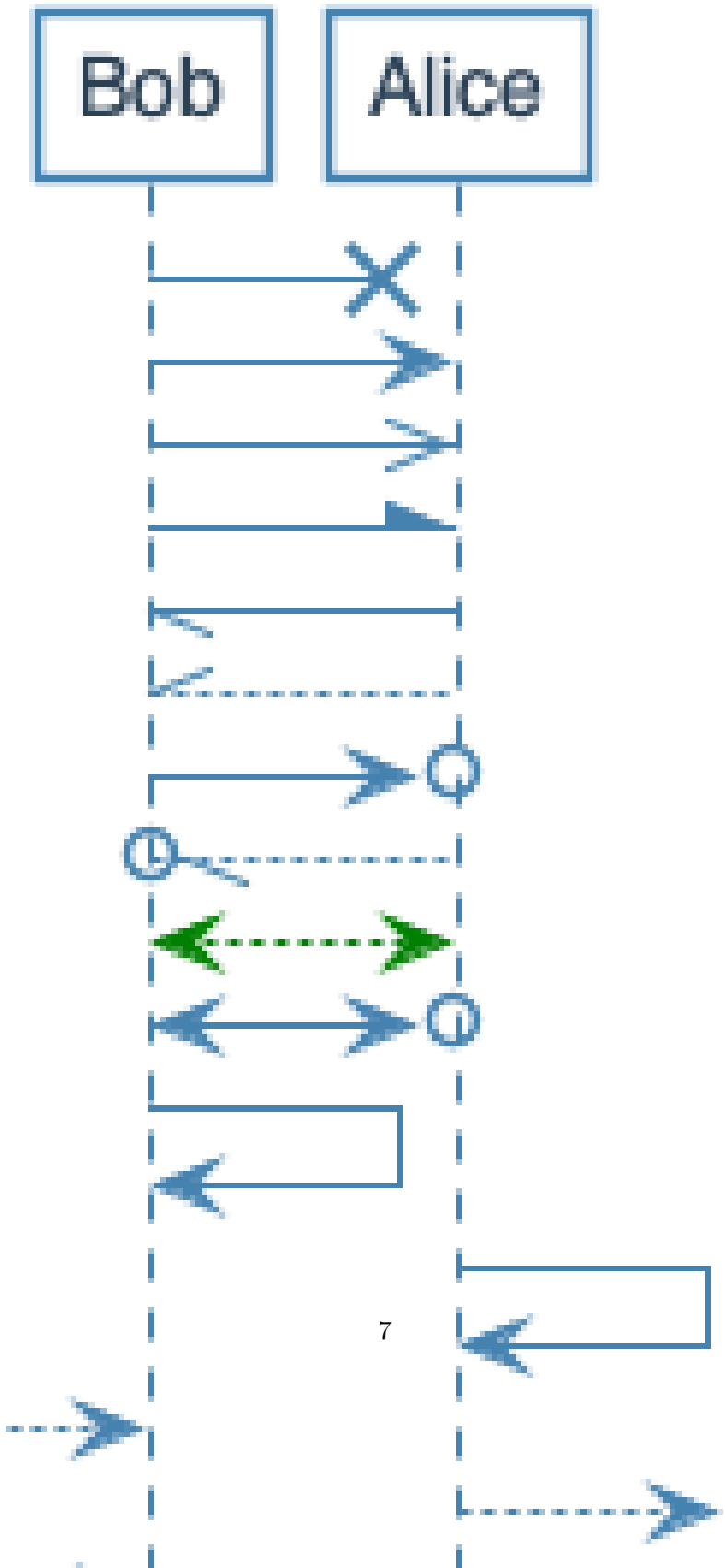
`->o` Arrow points to circle

`<->` Bidirectional arrow

`-[#red]->` Arrow color

`[->`, `->]`, `[<-`, `<-]` Un-anchored lines

4.1.1 Example Sequence Diagram Arrows



4.2 Participants

`actor ActorName` Define an actor

`boundary BoundaryName` Define boundary system

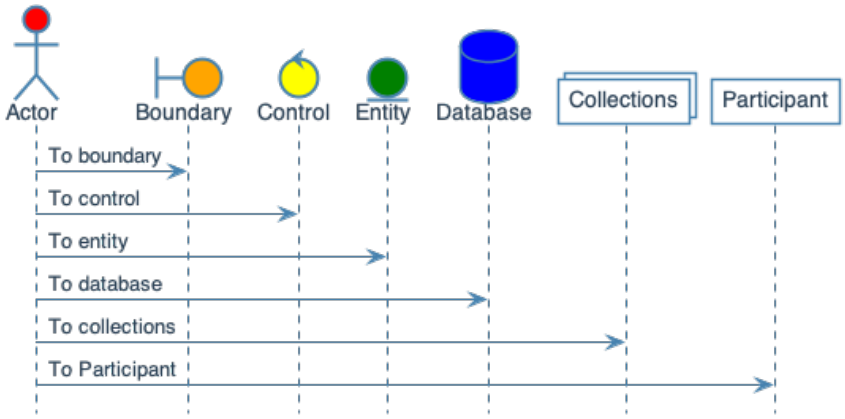
`control ControlName` Define a control system

`entity EntityName` Define an entity

`database DatabaseName` Define a database

`collections CollectionsName` Define a collection

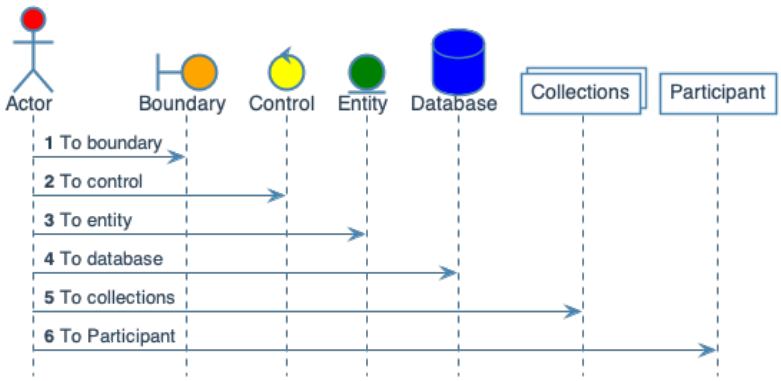
4.2.1 Example Sequence Diagram



4.2.2 Sequence Numbering

`autonumber <start>` Begin numbering at <start> or 1

1. Example autonumbering



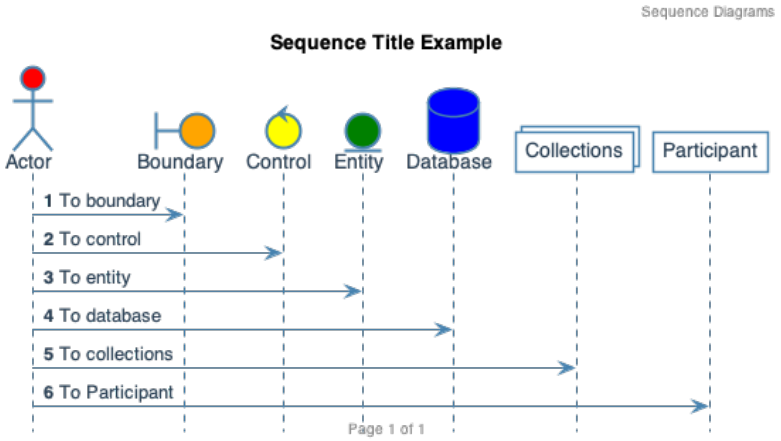
4.2.3 Page Titles, Headers, Footers

`header <content>` Defines header content

`title` <**content**> Defines title content

`footer <content>` Defines footer content

1. Example Titles, Headers, Footers



4.3 Sequence Grouping Boxes

`alt <title> ...  else <title> ...  end` Define alternates

`loop <title>` Steps in group are looped

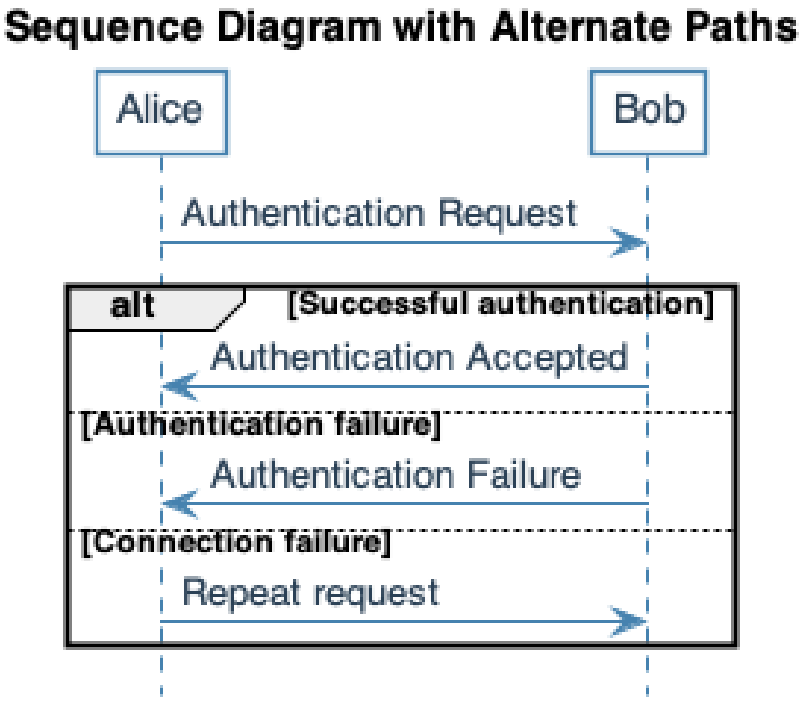
`par` Steps run in parallel

`break` Steps supercede subsequent steps

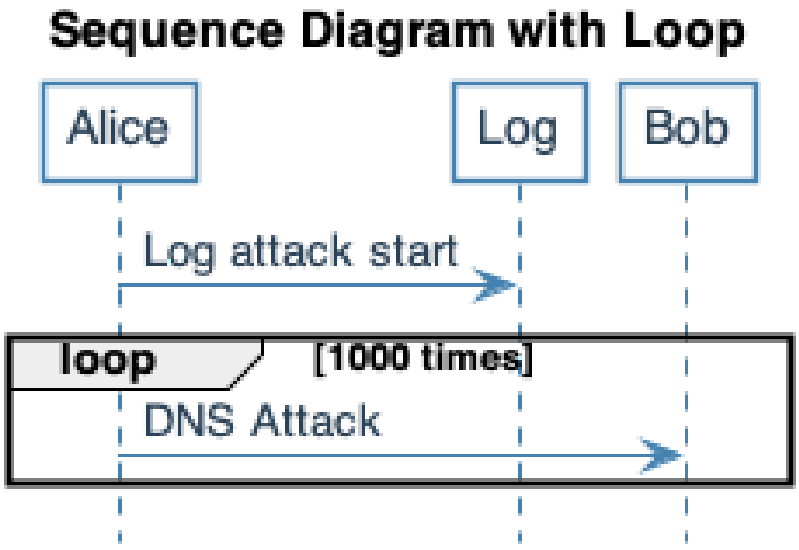
`critical` Steps must not be interleaved

`group <title>` Custom grouping

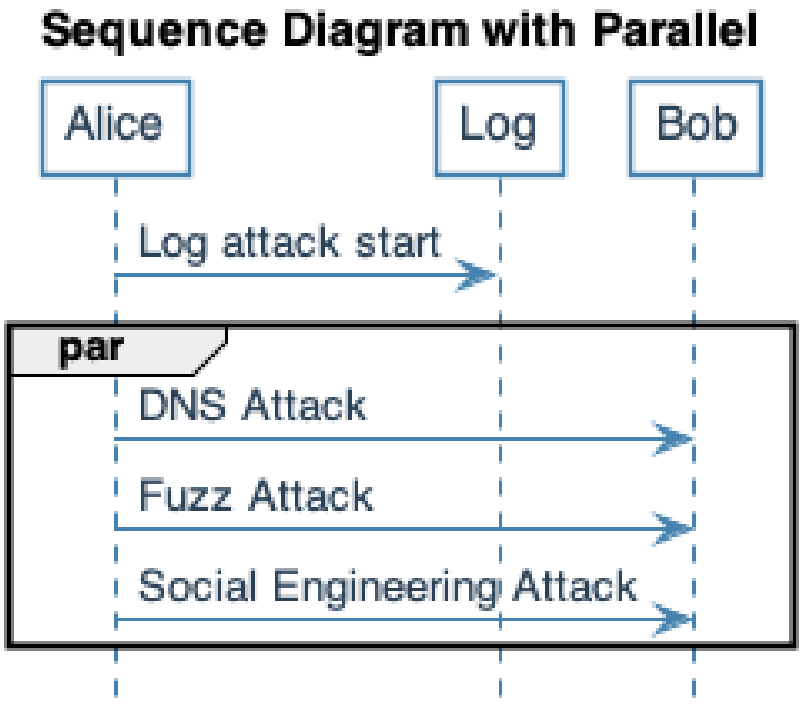
4.3.1 Example Sequence Diagram - Alternates



4.3.2 Example Sequence Diagram - Loops



4.3.3 Example Sequence Diagram - Parallel



4.4 Notes

4.4.1 Notes relative to message

Put note below the message

`note left: <note text>` Add a note to the left

`note right: <note right>` Add a note to the right

`note left ... end note` Multiline note

4.4.2 Notes relative to participant

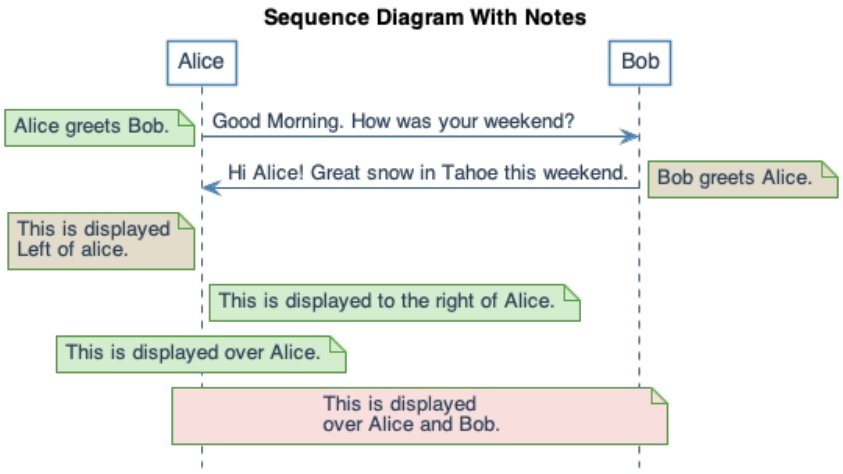
Put note below the participant

`note left of <Participant>` Place note left of participant

`note over <Participant>` Place note over participant

`note right of <Particpant>` Place note right of participant

4.4.3 Example Sequence Diagram With Notes



4.5 Formatting

`**bold**` Bold text

`//italics//` Italics text

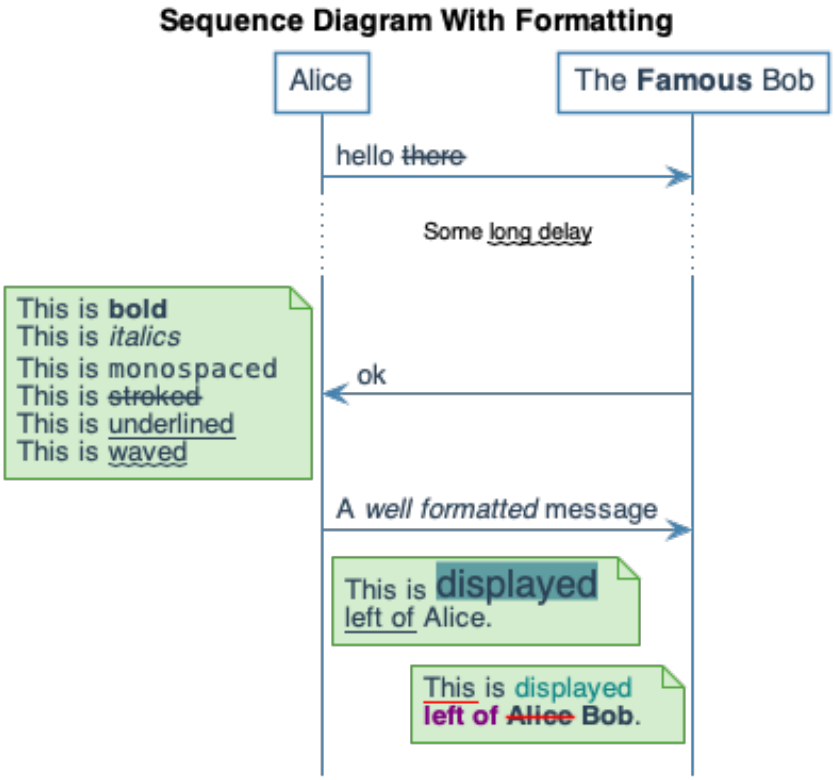
`""monospaced""` Monospaced text

`--strikethrough--` Strikethrough text

`__underline__` Underlined text

`~~wavy underline~~` Wavy underline text

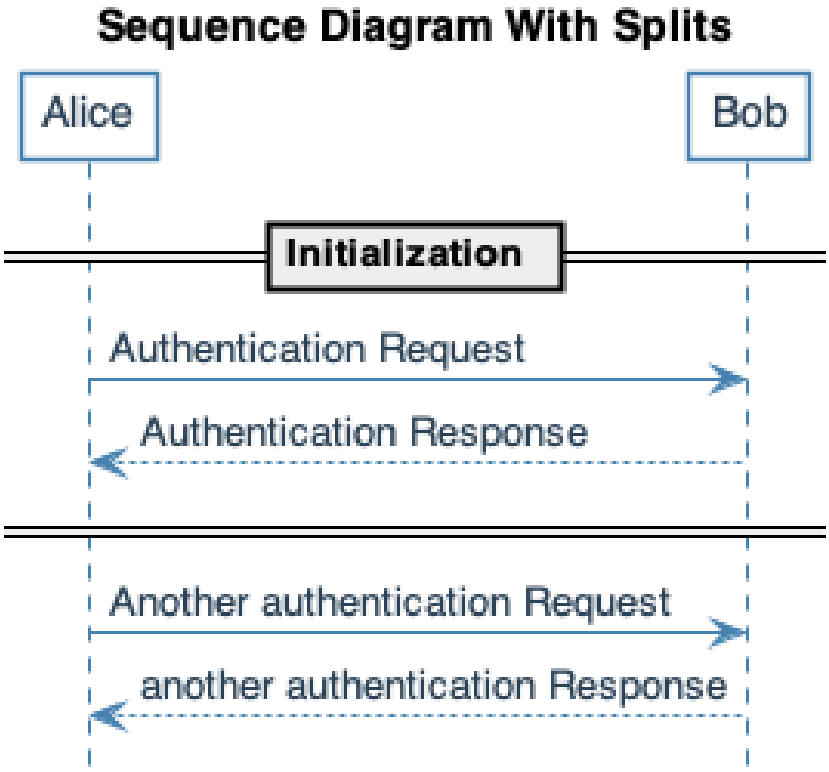
4.5.1 Example Sequence Diagram With Formatting



4.6 Splits

`== label ==` Insert a split

4.6.1 Example Sequence Diagram With Split

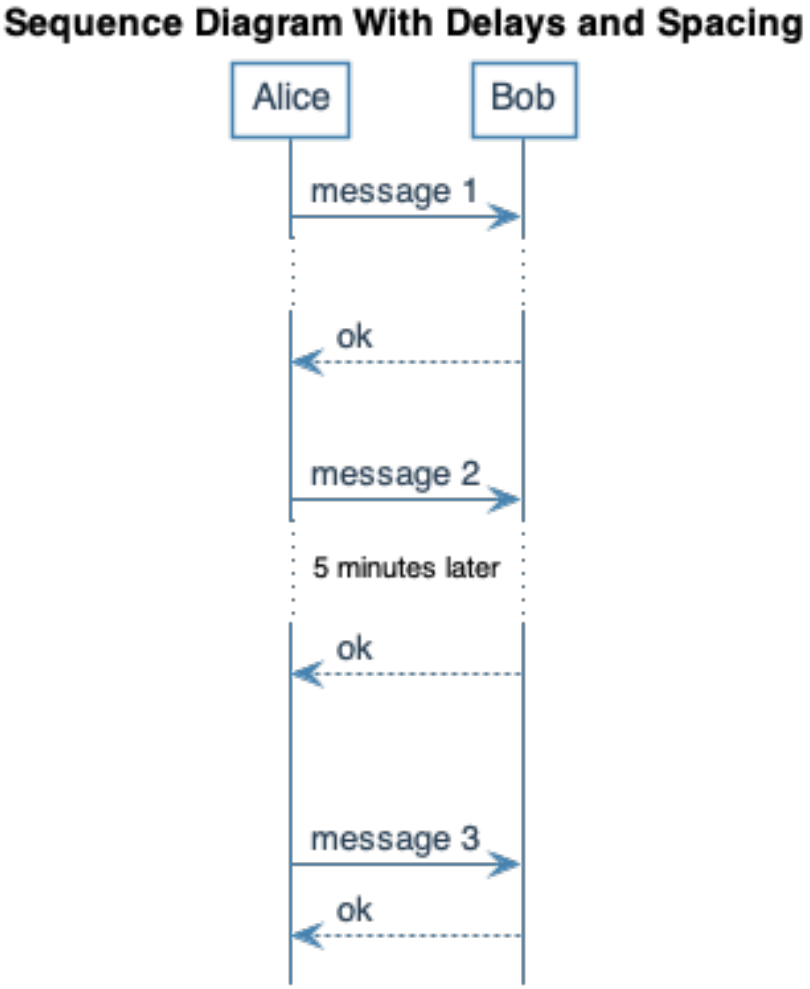


4.7 Delays & Spacing

. . . A delay in the sequence

| | | Add extra Vertical Spacing

4.7.1 Example Sequence Diagram With Delays And Spacing



4.8 Lifelines

`activate <id>` **or** `++` Mark start of lifeline <id>

`deactivate <id>` **or** `--` Mark end of lifeline <id>

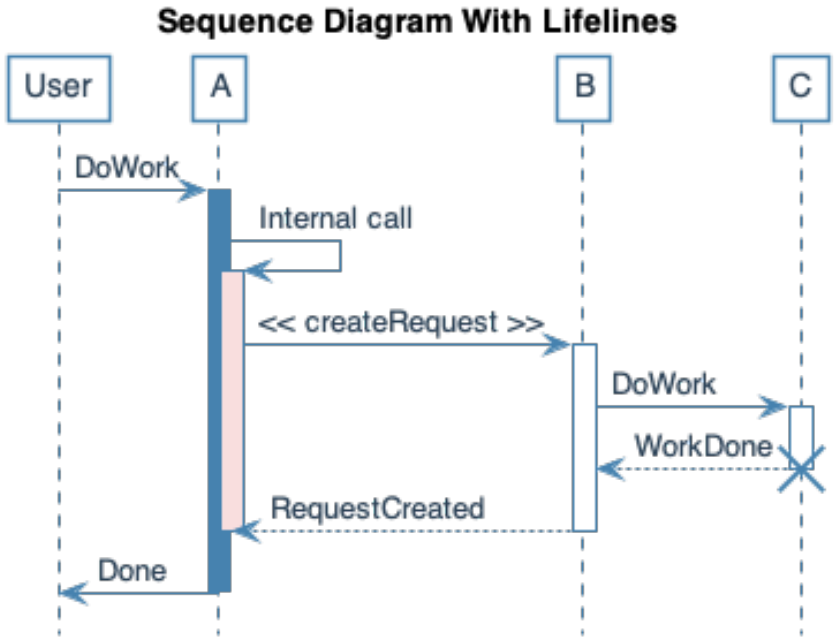
`autoactivate on` Turn on autoactivation

`create <type> <name>` **or** `**` Create a participant

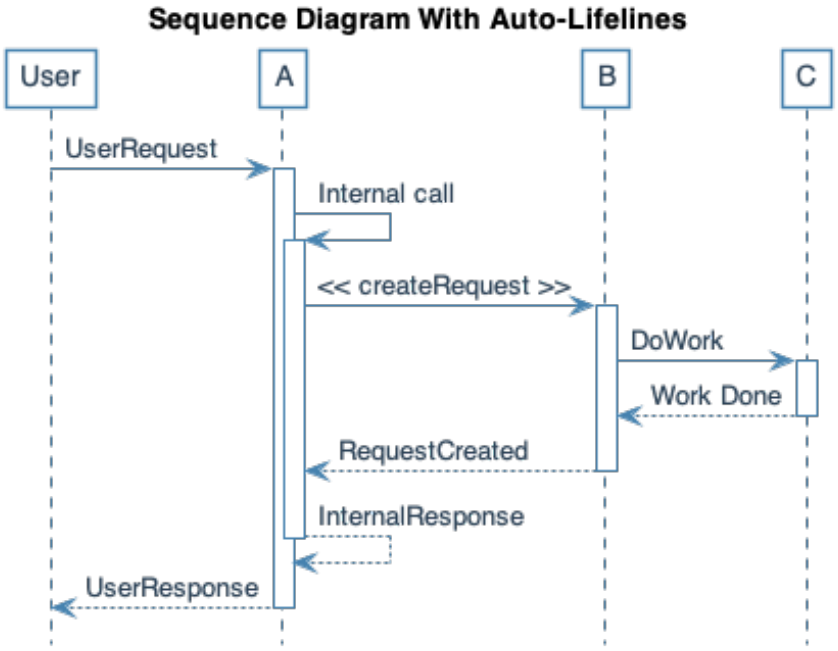
`destroy <id>` **or** `!!` Destroy participant

`return <label>` Deactivate lifeline with <label>

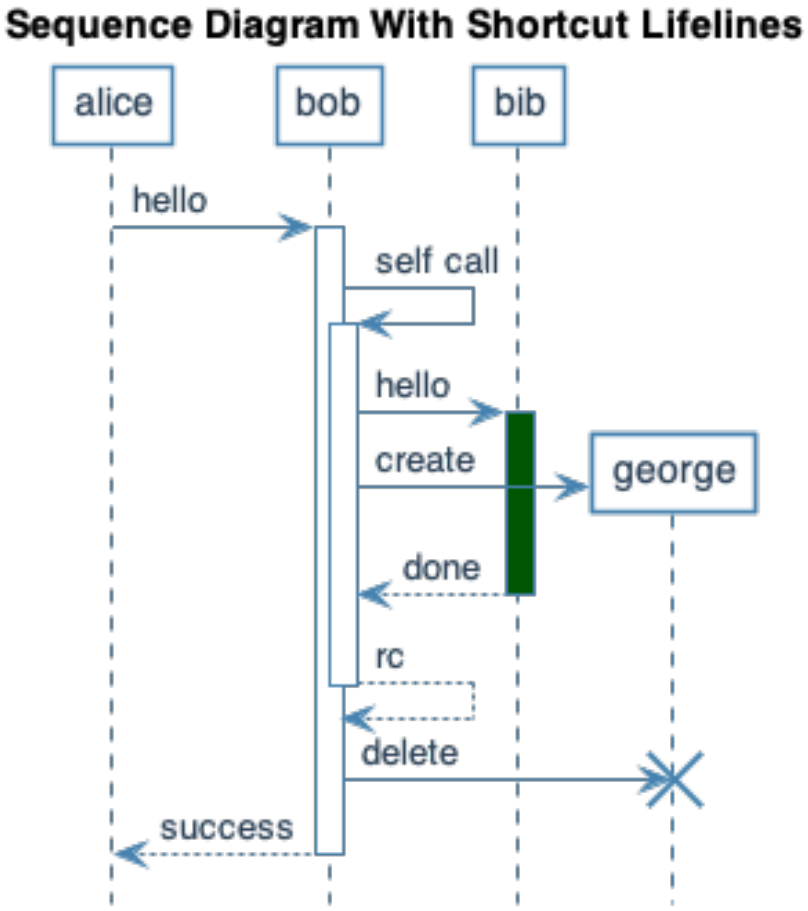
4.8.1 Example Sequence Diagram With Lifelines



4.8.2 Example Sequence Diagram With Auto-lifelines



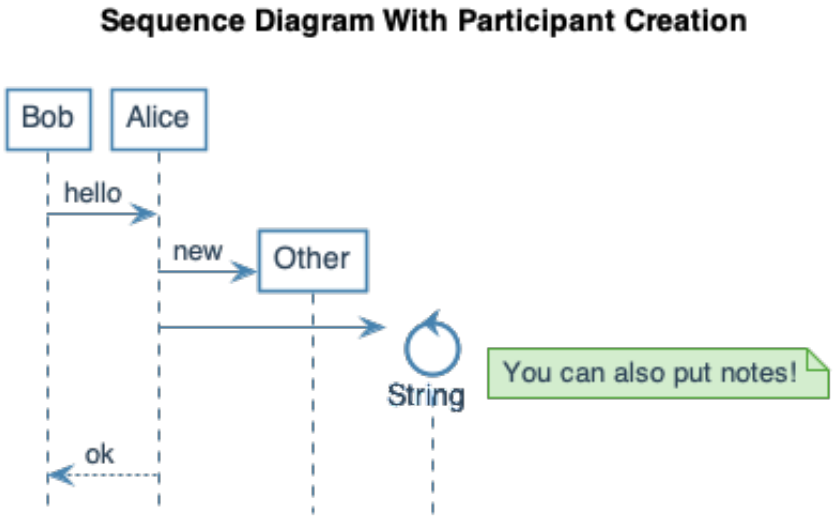
4.8.3 Example Sequence with Shortcut Lifelines



4.9 Participant Creation

~create <participant type> <participant name> Create a participant

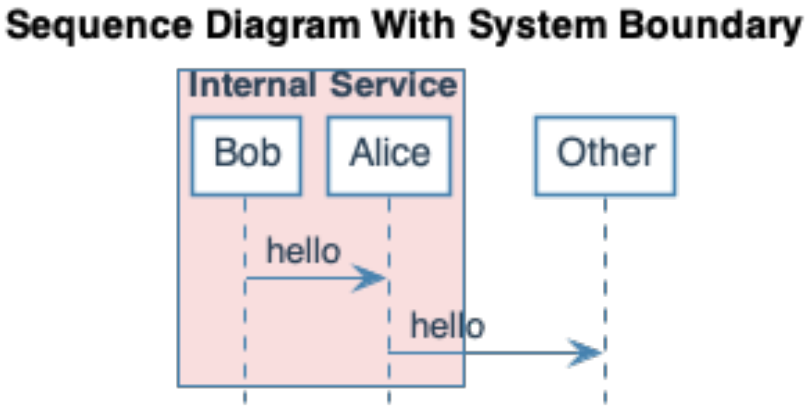
4.9.1 Example Sequence Diagram With Participant Creation



4.10 System Boundaries

`box <name> <color> ...  end box` Wrap in a box

4.10.1 Example Sequence Diagram With System Boundary



5 Entity Diagrams

5.1 Entities

`entity EntityName` Define an entity

`* indentifying_attribute` Identifying attribute

`--` Divider

`* indentifying_attribute` Identifying attribute

`optional_attribute` Optional attribute

5.2 Connectors

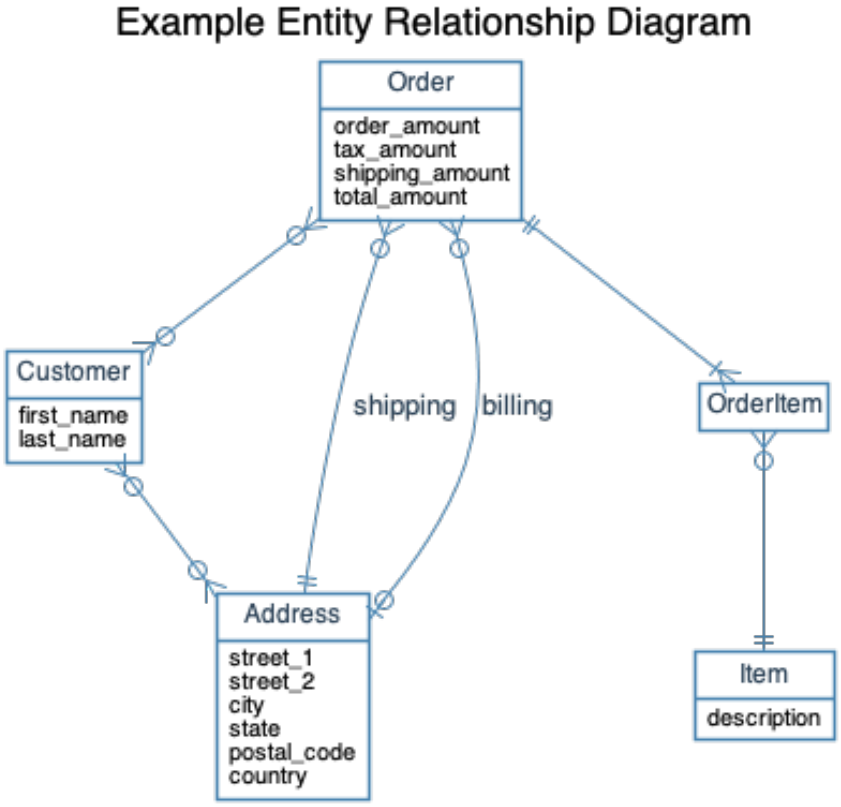
`A |o--o| B` Zero or one

`A ||--|| B` Eactly one

`A }o--o{ B` Zero or many ("crows foot")

`A }|--|{` One or many

5.3 Example Entity Diagram



* Class Diagrams

5.4 Lines

`--` Solid line

`..` Dotted line

5.5 Arrows

`<|--`, `--|>` Extension

`*--`, `--*` Composition

`o--`, `--o` Aggregation

`()--`, `--()` Lollipop (interfaces)

5.6 Vertical vs Horizontal Layout

`--` Relationships with two dashes are laid out vertically

`-` Relationships with one dash are laid out horizontally

5.7 Relations

`Class A -- Class B` Relation

`ClA -- ClB : label` Labeled relation

`Driver - Car : drives > <, >` "acts on" label

5.8 Association Class

`(Student, Course) .. Enrollment` Enrollment associates Student & Course

5.9 Fields & Methods

`<ClassName> : <field>` Add a field to a class

`<ClassName> : <method>()` Add a method to a class

`class <ClassName> { (fields/methods) }` Define a class

5.10 Visibility

- `-` : private
- `#` : protected
- `~` : package private
- `+` : public

5.11 Abstract, Static, Stereotypes, annotation, enum

`{static}` static modifier

`{abstract}` Abstract modifer

`« name »` Stereotype (interface)

`enum` Define an enumeration

anootation Define an annotation

5.12 Separators

`==` Thick line

`..` Dotted line

`--` Plain line

`__` Medium line

`-- label --` Labeled line

5.13 Notes

`note "text" as N1` Define a floating note; attach with `..`

`note left of <class>` Attach note above <class>

`note right of <class>` Attach note right <class>

`note top of <class>` Attach note top <class>

`note bottom of <class>` Attach note bottom <class>

`note <left|right|top|bottom> on link` Add a note to a link

5.14 Example Class Diagram