Diagramming in Software engineering:
SE Radio episode 566

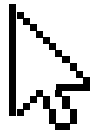
Gonzalo Alonso Fernández UO282104

Raúl Mera Soto UO287827

Laura Gómez Menéndez UO275725

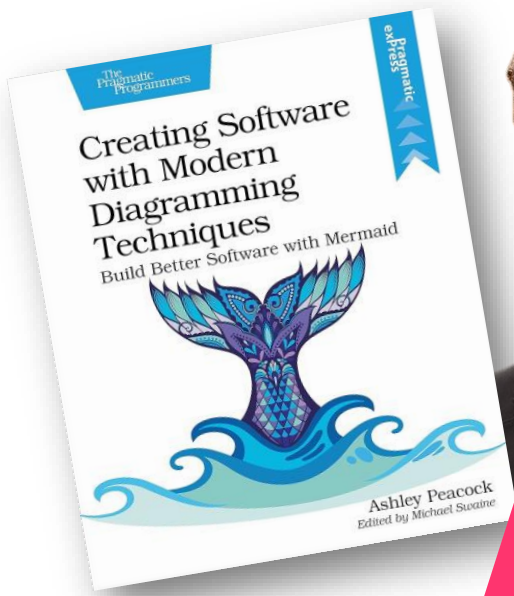
Daniel Sinne Argüelles UO282500

Table of contents



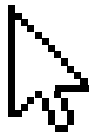
SOFTWARE ENGINEERING RADIO PODCAST

Introduction



Why is diagramming important?

- Make information digestible, providing a visual and easy-to-understand representation.
- Essential for documenting architectures and explaining complex concepts.
- Can be applied in the whole software life cycle.



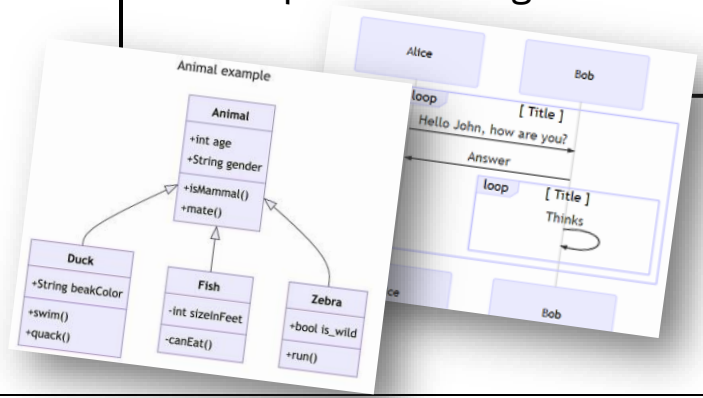
Ashley's opinion about UML

- UML's poor reputation.
- Emphasis on Notation over Diagram Power.
- Advancements in diagramming tools.
- Lack of Awareness and Educational Resources.

Types of Diagrams in Software Engineering

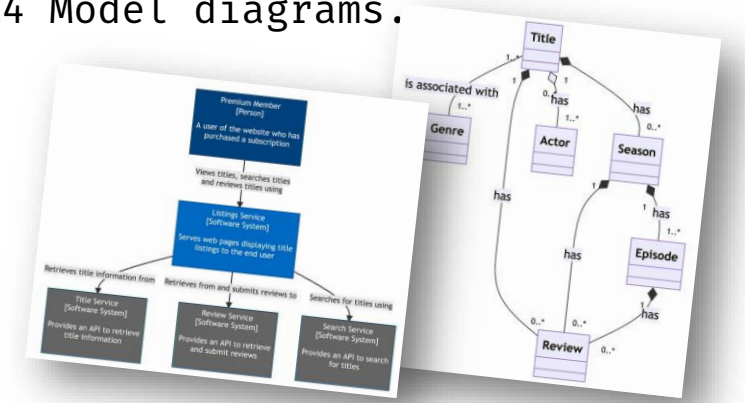
Short-lived diagrams

- Class diagrams.
- Sequence diagrams.



Long-lived diagrams

- Domain Model diagrams.
- C4 Model diagrams.



Tools for modern diagramming techniques

Draw.io

Ease of use

PlantUML

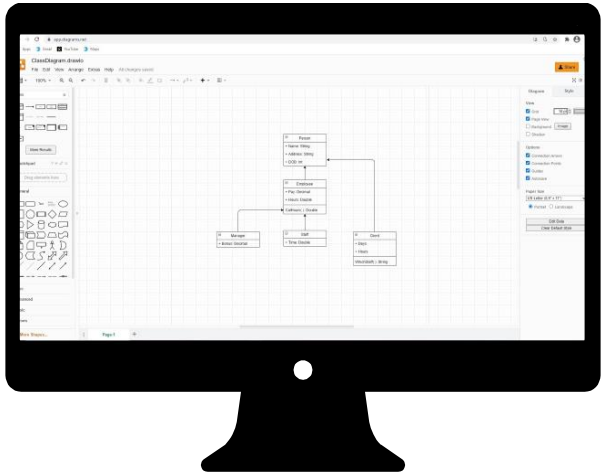
Diagrams as code

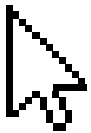
Mermaid

Browser rendering

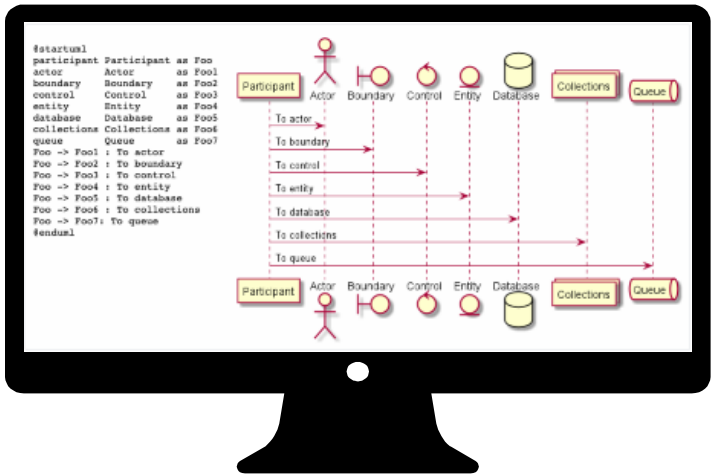
Draw.io

+ Open source
+ Easy to use (Drag & drop)
- Slow to change
- Maintenance nightmare
- Unreadable format
- Bad option for version control





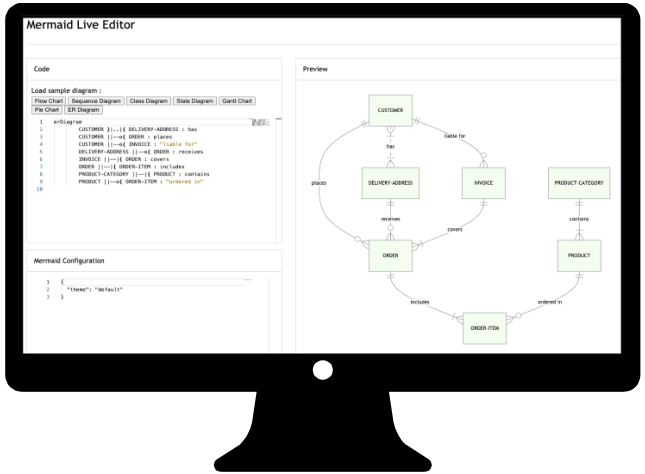
PlantUML



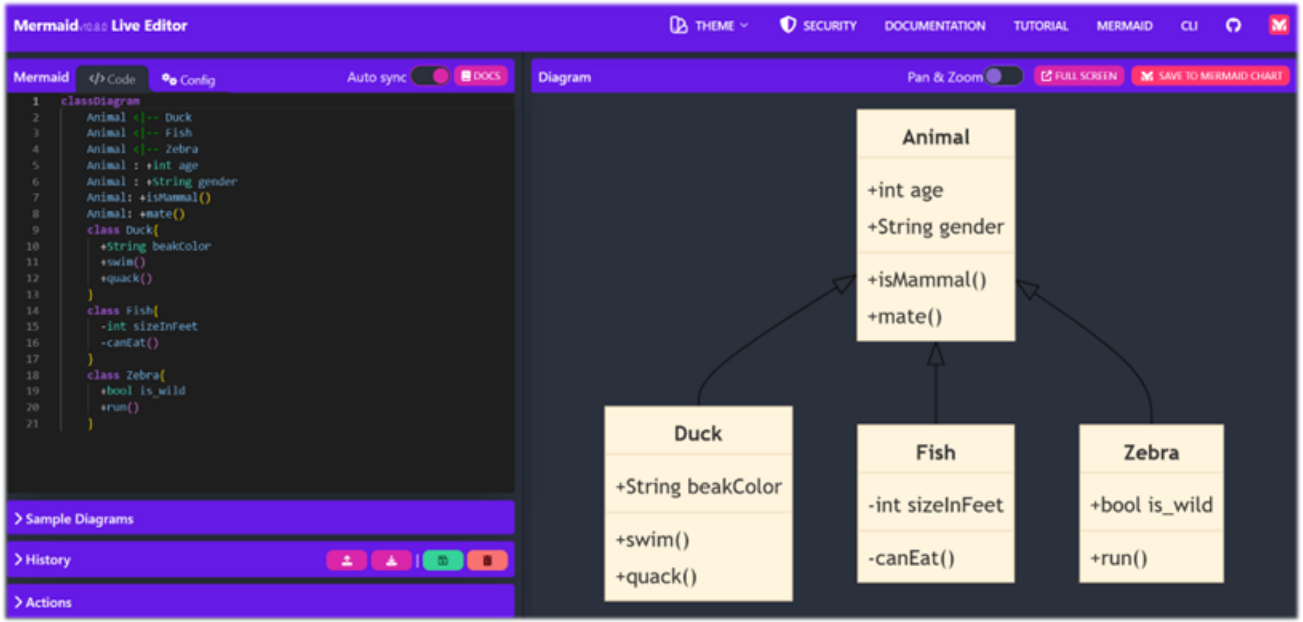
+ Open source
+ Diagrams as code
+ Text-based
  + Fast creation and editing
  + Consistency
  + Version control friendly
- Needs Java

Mermaid

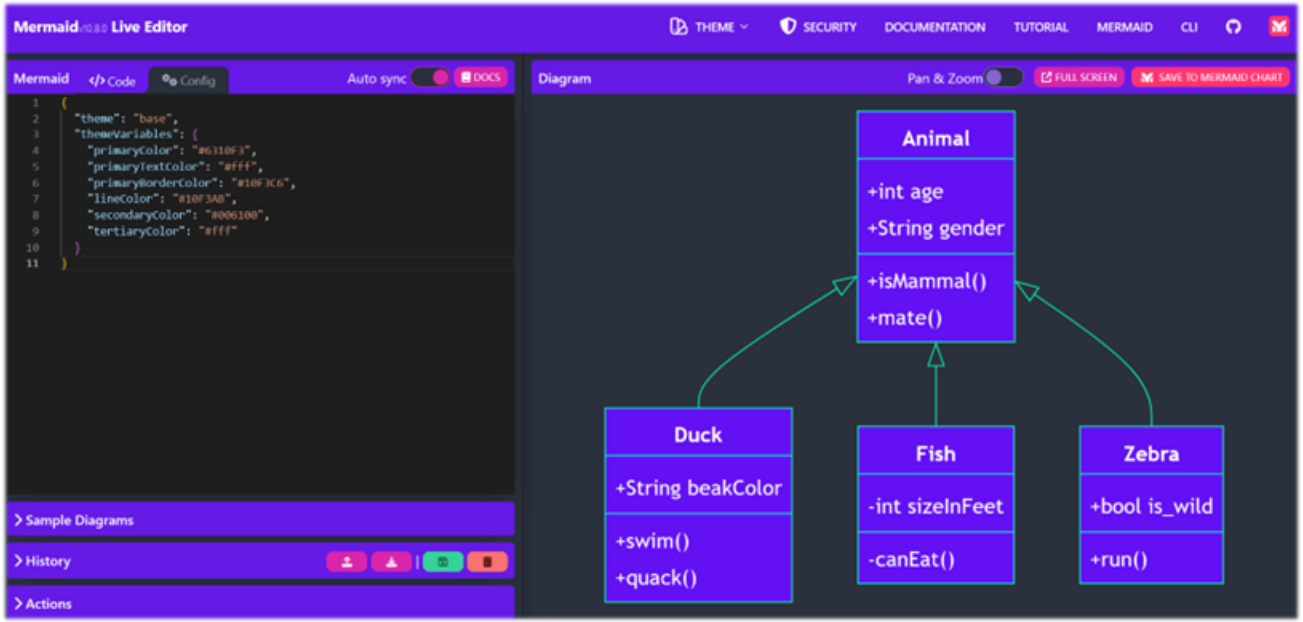
+ Open source
+ Similar to PlantUML
  + JavaScript
  + No separate diagram tool
  + Browser rendering
  + Easier version control



Mermaid syntax



Mermaid styling



Mermaid Sequence Diagrams

Handle the layout and the rendering for you

Lifelines

Individual nodes the messages are passing between

Flowcharts

Square boxes, cylinders, diamonds...

Dotted lineback

To show responses

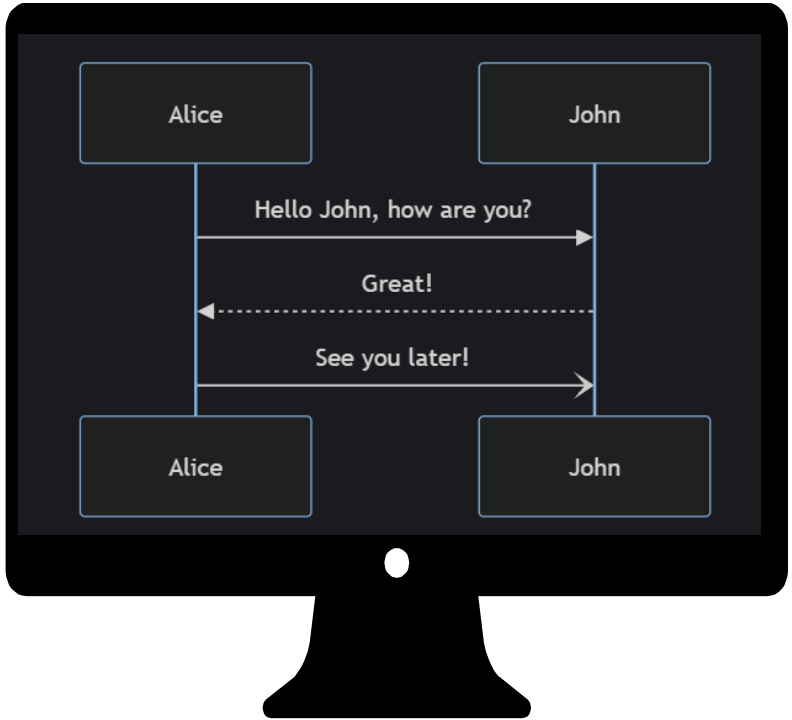
Solid line

To show requests

Mermaid Sequence Diagrams

Participants

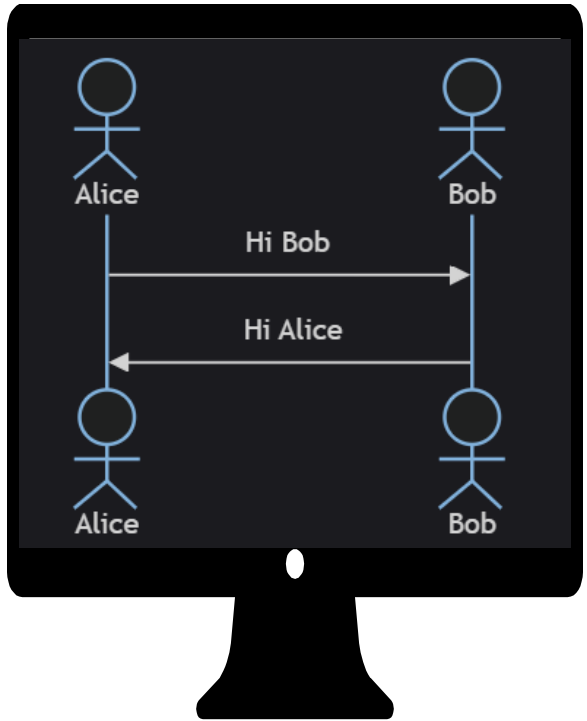
```
sequenceDiagram
 Alice->>John: Hello John,how are you?
 John-->>Alice: Great!
Alice-)John: See you later!
```



Mermaid Sequence Diagrams

Actors

```
sequenceDiagram
    actor Alice
    actor Bob
    Alice->>Bob: Hi Bob
    Bob->>Alice: Hi Alice
```



C4 Model

Simon Brown created this way to model your software Architecture.

The name C4 Model comes from 4 diagrams.

C4 Model

Diagrams

System Context Diagram

Highest level of abstraction and provides an overview of the system

Container View

Show how the containers interact with each other and with external systems or users

Component Diagram

Explores deep the internal structure of each container

Code diagram

Provides a detailed view of the code-level elements within the components

Component vs Container

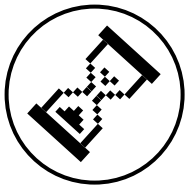
Represent different architectural elements in a system

Component

Specific software module

Container

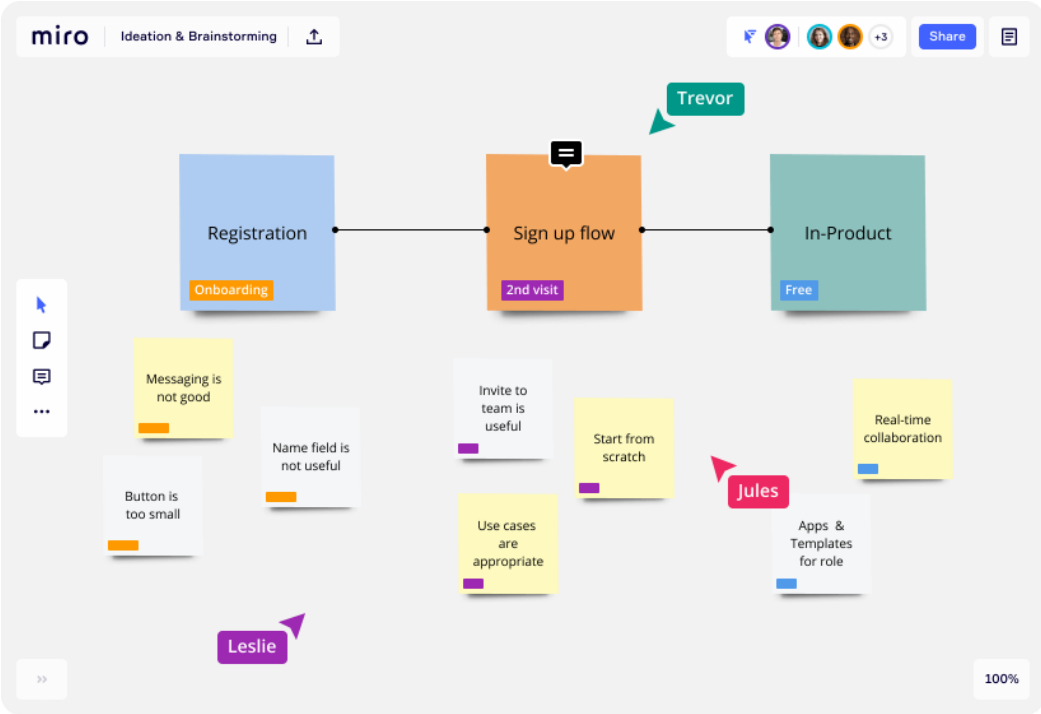
Higher-level grouping or environment that contains multiple components or other containers.



Other Useful Tools

Miro

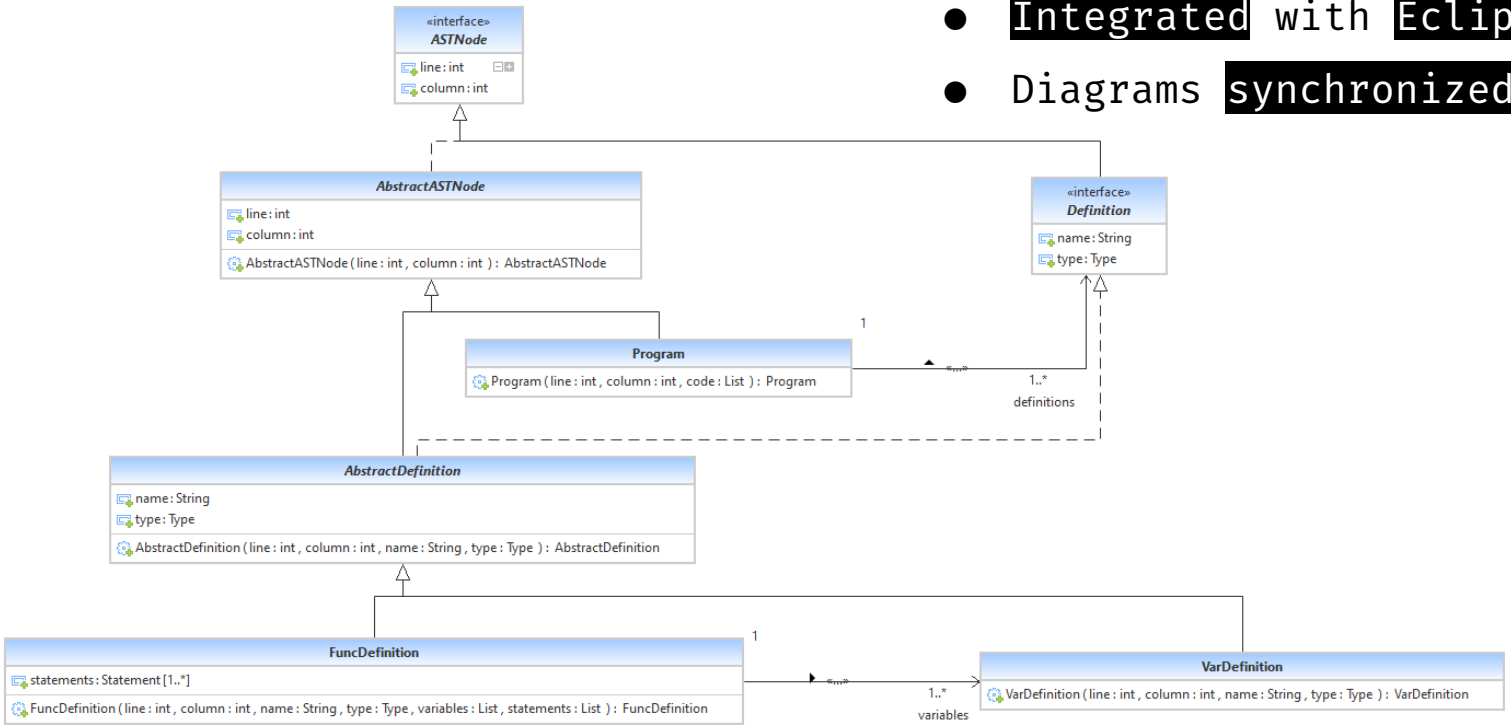
- Digital whiteboard
- Used for brainstorming and event storming (Domain-Driven Design)
- Work remotely



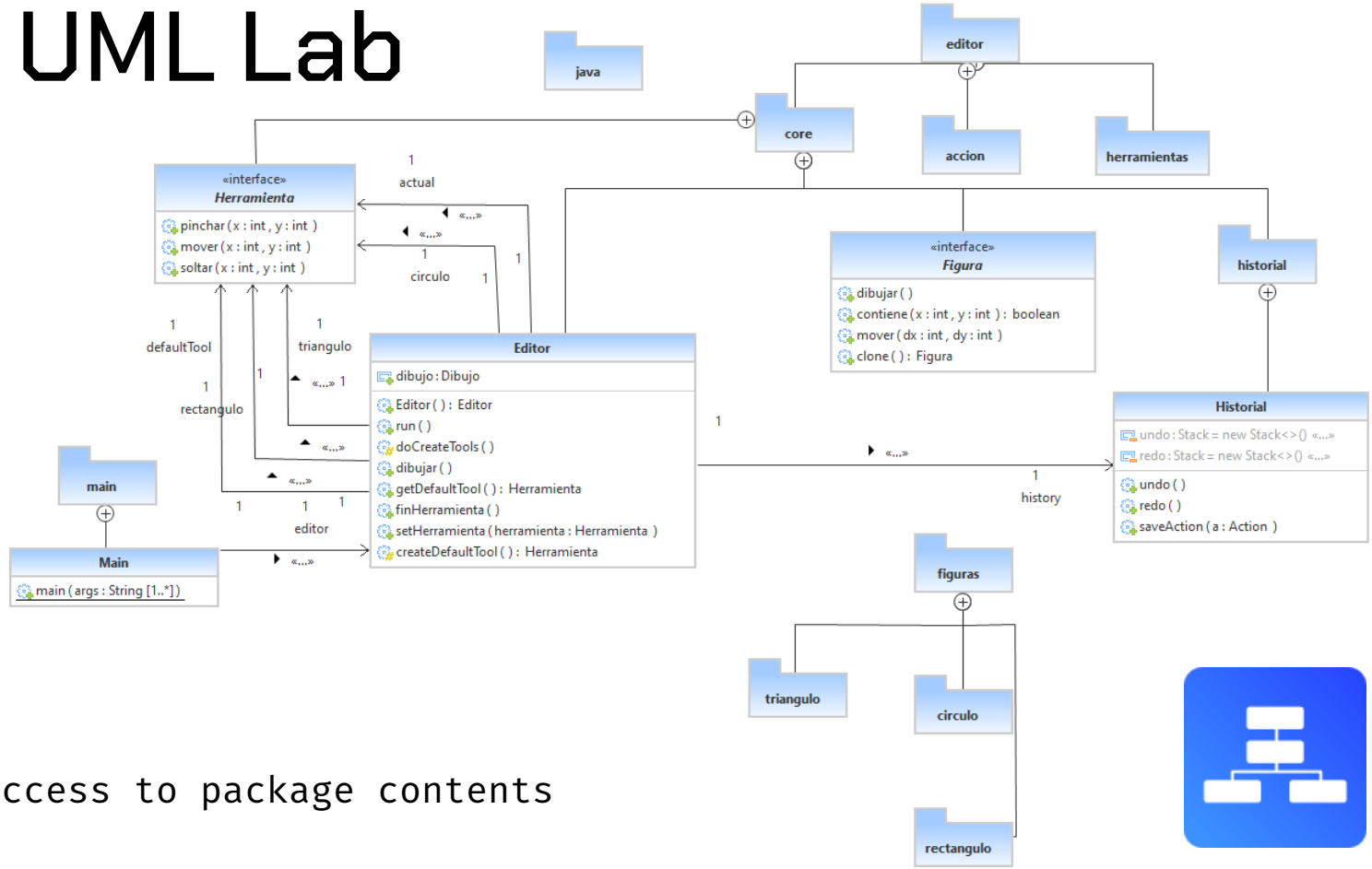
miro

UML Lab

- Integrated with Eclipse IDE
- Diagrams synchronized with code



UML Lab



- Dynamic access to package contents

Any question?

Thank you