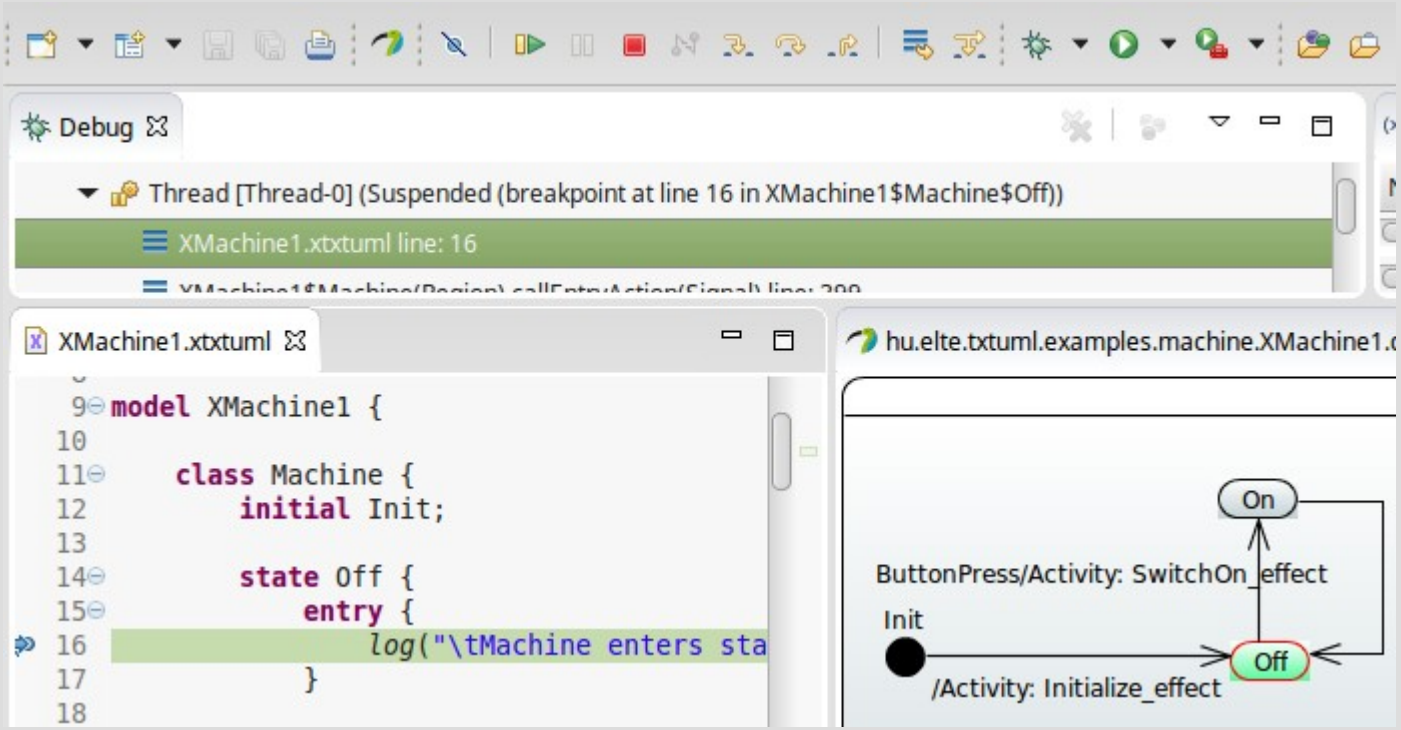
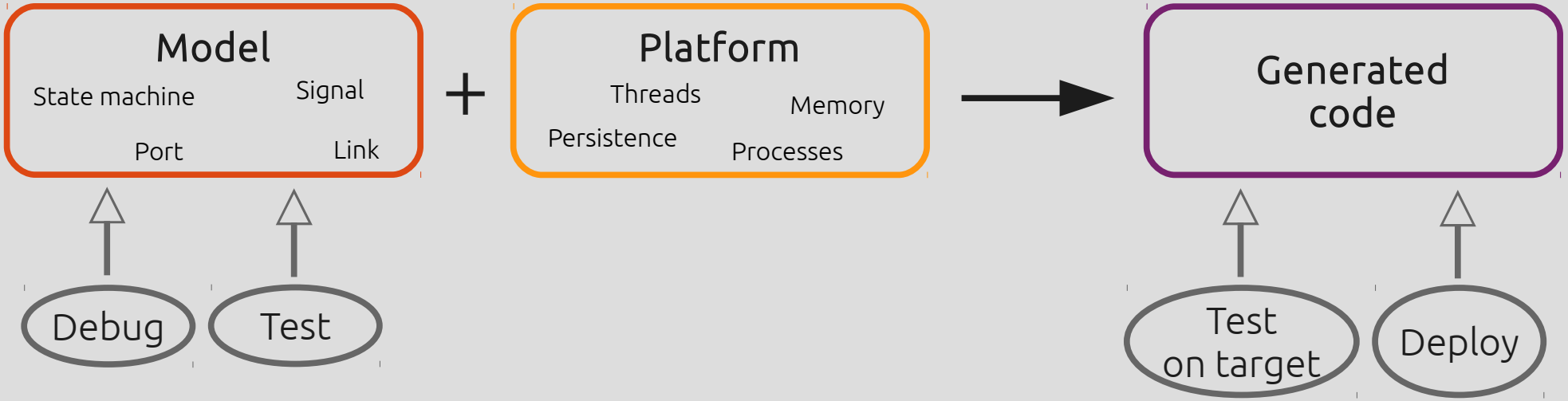
Advances in UML Model Execution



Ericsson Modeling Days
Nov 16, 2015

Gergely Dévai
ELTE-Soft Ltd.

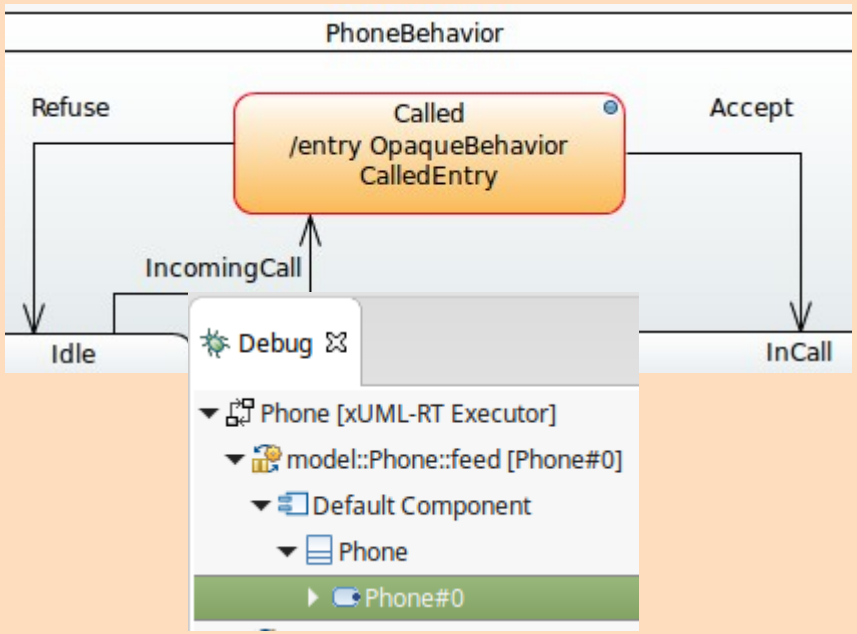
Executable Modeling Vision



- Early verification
- Avoid model - code divergence
- Separation of business logic and platform-specifics
- Predictable code quality

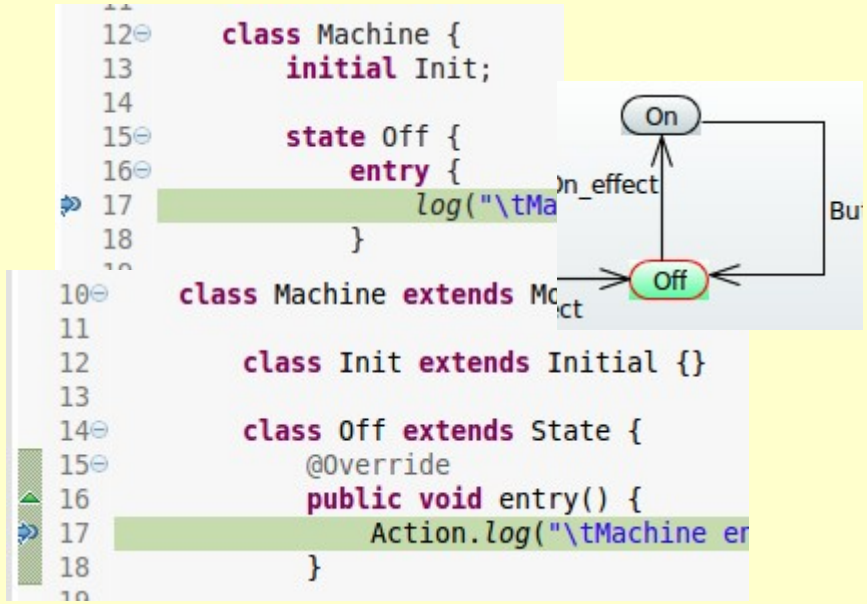
Two Approaches

Model execution via code generation



xUML-RT Model Executor

Text-based model execution



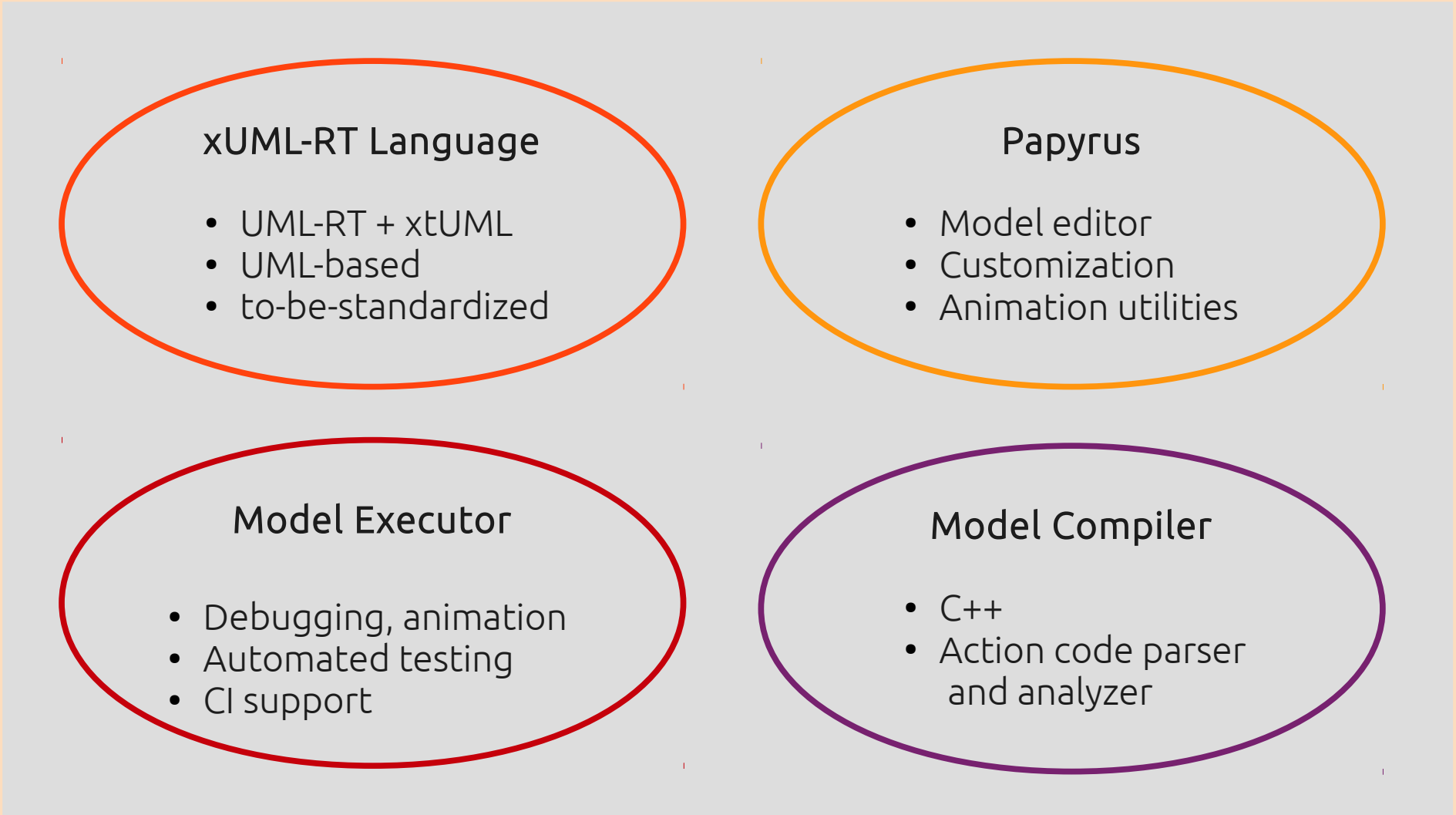
txtUML Framework

Model Execution via Code Generation

xUML-RT Model Executor

xUML-RT Model Executor

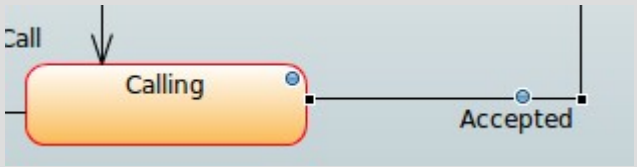
Project Background

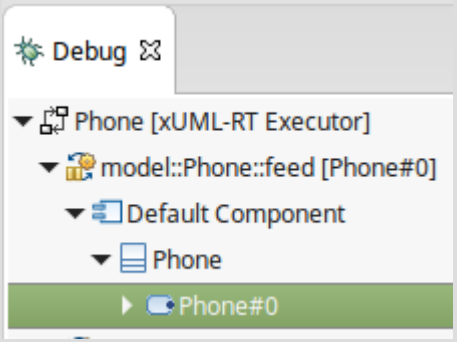


xUML-RT Model Executor

Goals

- Interactive model execution and debugging
  - Breakpoints in action code and on diagrams
  - Debug controls
  - Animation of state machines
  - View instances and their properties

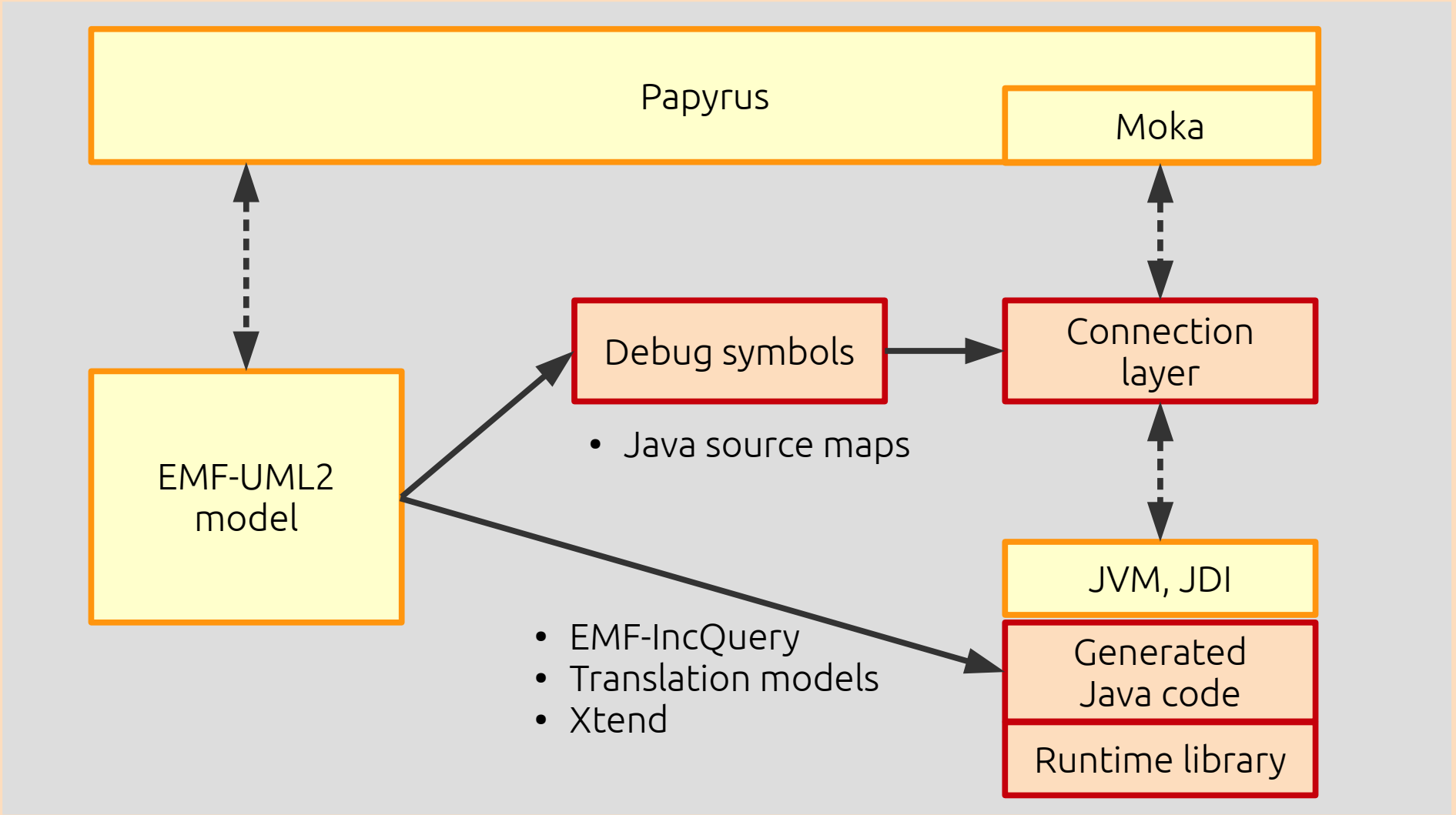




- Mass test execution on model level
  - Runtime performance is important
  - Randomization of non-deterministic choices
  - Log-replay functionality
- Support for class and capsule modeling

xUML-RT Model Executor

Model Execution via Code Generation



xUML-RT Model Executor

Demo

Lessons Learned

- The code generation approach works
  - Provides good performance for testing
  - Debugging and animation is possible
- Tricky parts:
  - UI programming
  - Integration of technologies
- Being part of a work-in-progress tool-chain…
  - Customized editor, Model refactoring, Search
  - Compare & merge, Model review etc.

Text-based Model Execution

txtUML Framework

txtUML Framework

Motivation

- Text input is faster than graphical for many users
- Tooling for edit, search, review, compare&merge is easier for textual languages
- Tooling for mainstream languages, like Java, is mature

➔ Idea: *Implement the executable UML abstractions in text and reuse tooling*

- txtUML: **t**extual, e**x**ecutable, **t**ranslatable **UML**

{txtUML}

textual, executable, translatable

txtUML Framework

Features

- Two text front-ends:
  - Custom syntax
  - Java API

```
class Machine {
    initial Init;

    state Off {
        entry {
            log("\tMac
        }
```

```
class Machine extends ModelClass {

    class Init extends Initial {}

    class Off extends State {
        @Override
        public void entry() {
            Action.log("\tMachine e
        }
```

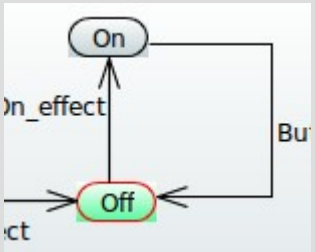
- Validation of modeling rules

```
Vector<User> users;
```

- Debugging, breakpoint support

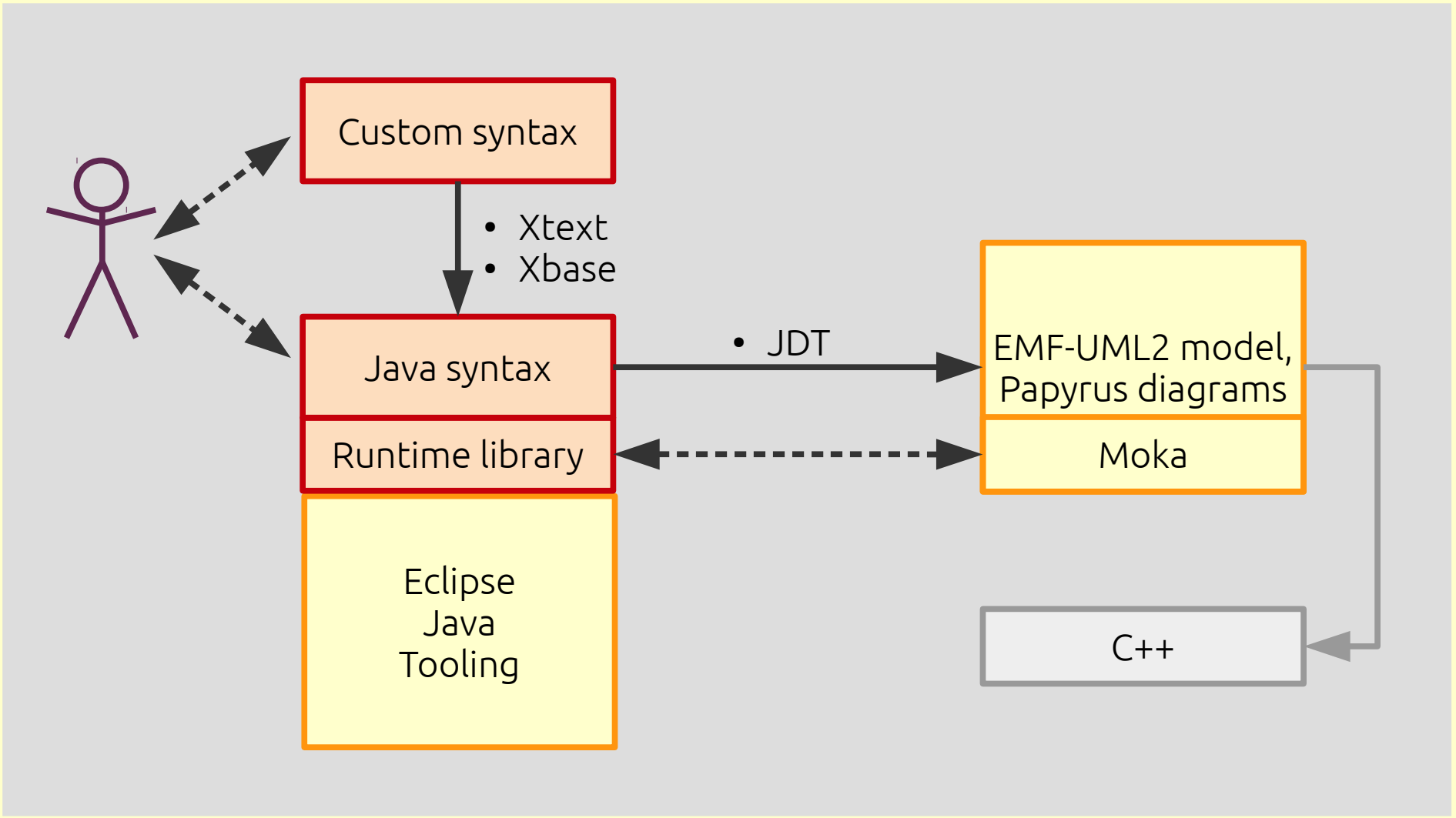
```
12    class Machine {
13        initial Init;
14
15        state Off {
16            entry {
17                log("\tMa
18            }
19
```

- Papyrus diagram generation
- Animation of generated state machines



txtUML Framework

Text-based Model Execution



txtUML Framework

Demo

txtUML Framework

Lessons Learned

- Lighter weight tool-chain
- Execution and debugging is (almost) for free
- Good runtime performance for testing
- Xtext, Xbase scalability:
  - Many files are handled well
  - Large files can be problematic
- Java limitations
  - Multiple inheritance

Thank you!

xUML-RT Model Executor
http://modelexecution.eltesoft.hu

txtUML Framework
http://txtuml.inf.elte.hu