

# txtUML

Model-driven Development Research Group,  
Eötvös Loránd University, Budapest

## Pragmatic approach to executable UML modeling

### Textual

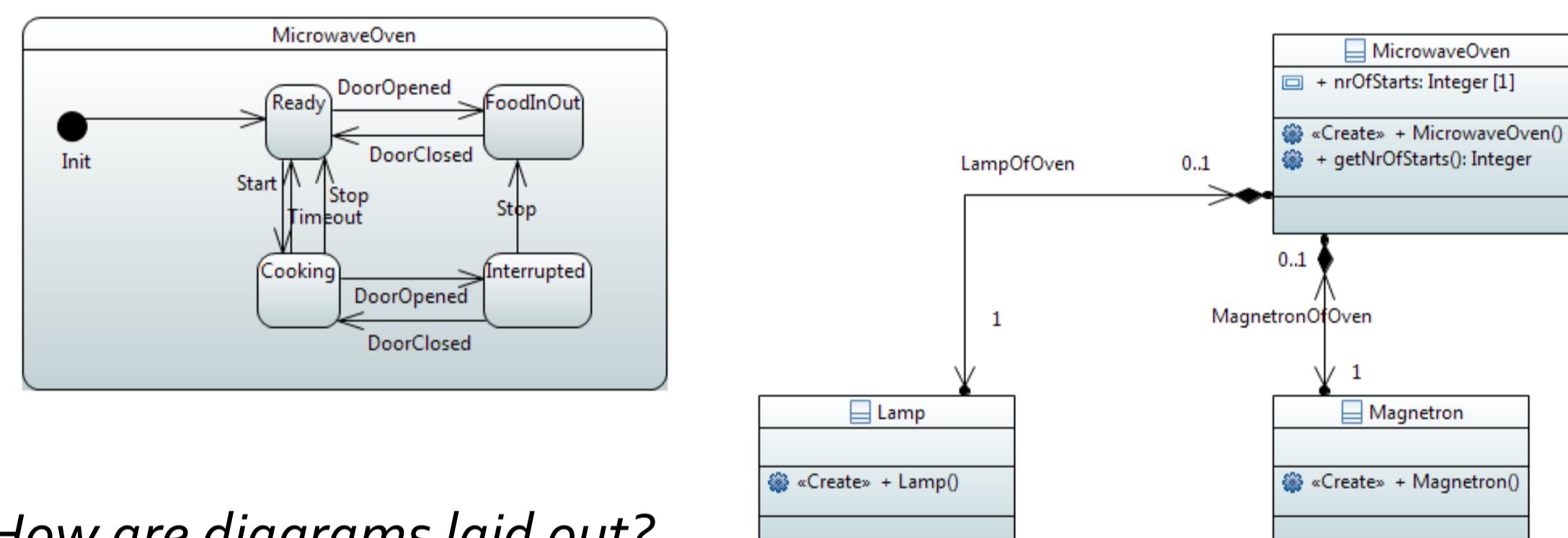
Models are edited in text,  
generated graphical diagrams help understanding.

```
*MicrowaveOven.txtuml
118 signal LampOn;
119 signal LampOff;
120
121 class Lamp {
122     initial Init;
123     state Off;
124     state On;
125 }
126
127 transition Initialize {
128     from Init;
129     to On;
130     trigger LampOn;
131     guard Off -> example.model.Lamp.On;
132     effect On -> example.model.Lamp.On;
133     guard to On -> Ctrl-Space to show shortest proposals;
134     guard trigger LampOn;
135 }
136
137 transition SwitchOff {
138     from On;
139     to Off;
140     trigger LampOff;
141 }
142
143 composition LampOfOven {
144     container MicrowaveOven oven;
145     1 Lamp lamp;
146 }
147
148 }
```

#### Why text?

- Mature editors
- Easier version control
- Advanced compare & merge tools
- Usually faster than editing graphics

#### Generated class and state machine diagrams

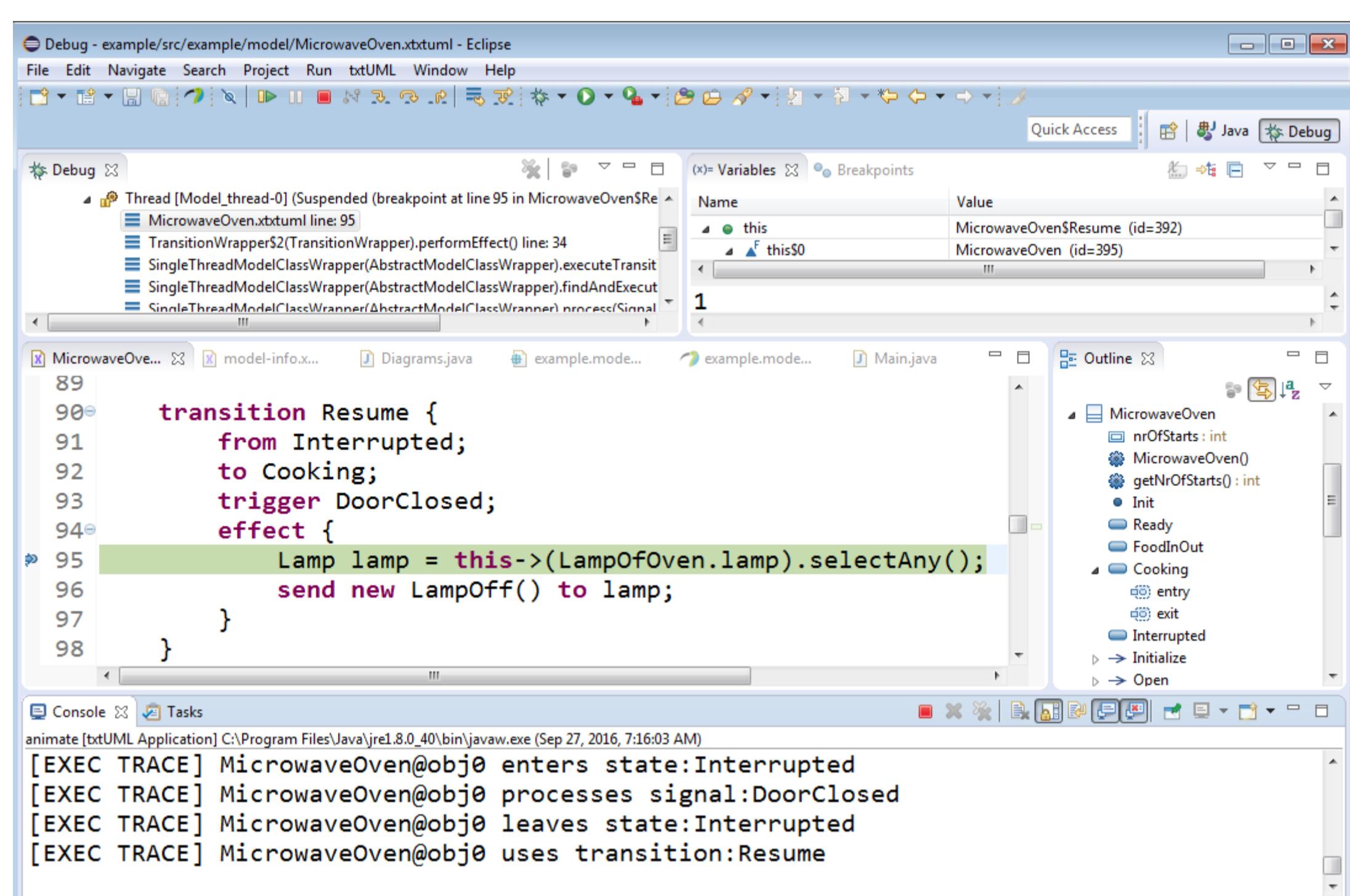


#### How are diagrams laid out?

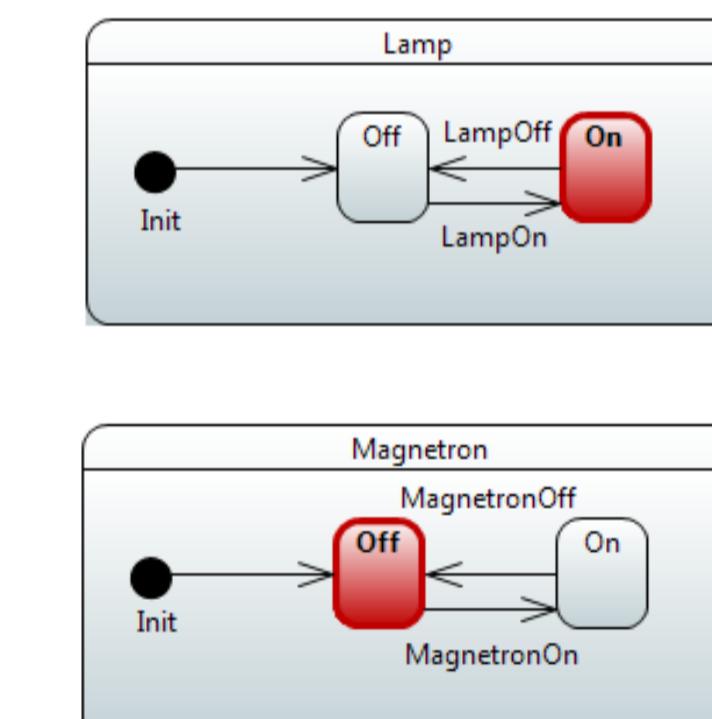
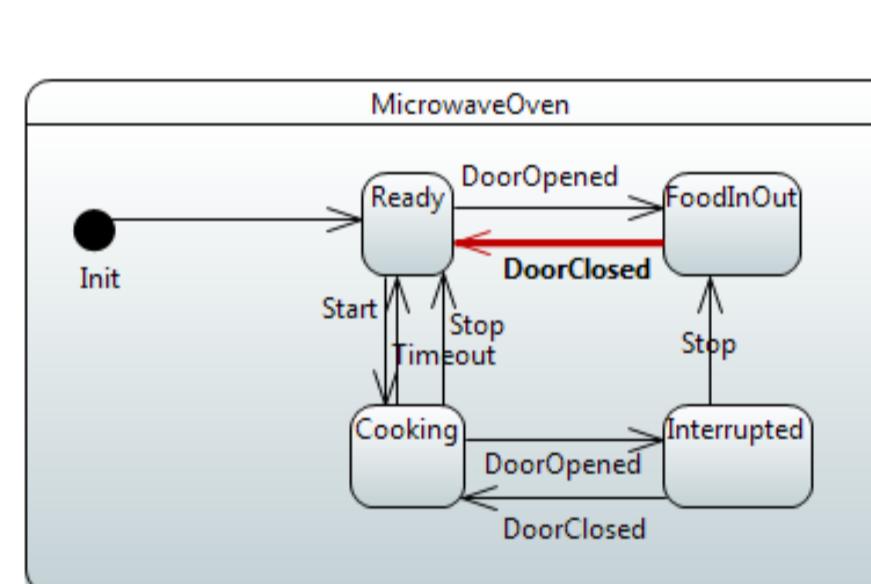
- Smart layout algorithm based on user constraints
- Constraints can be partial
- Constraints specified by simple textual descriptions
- Easy to version control

### Executable

Models can be executed, debugged, tested in Eclipse,  
and seamlessly integrated with Java software.



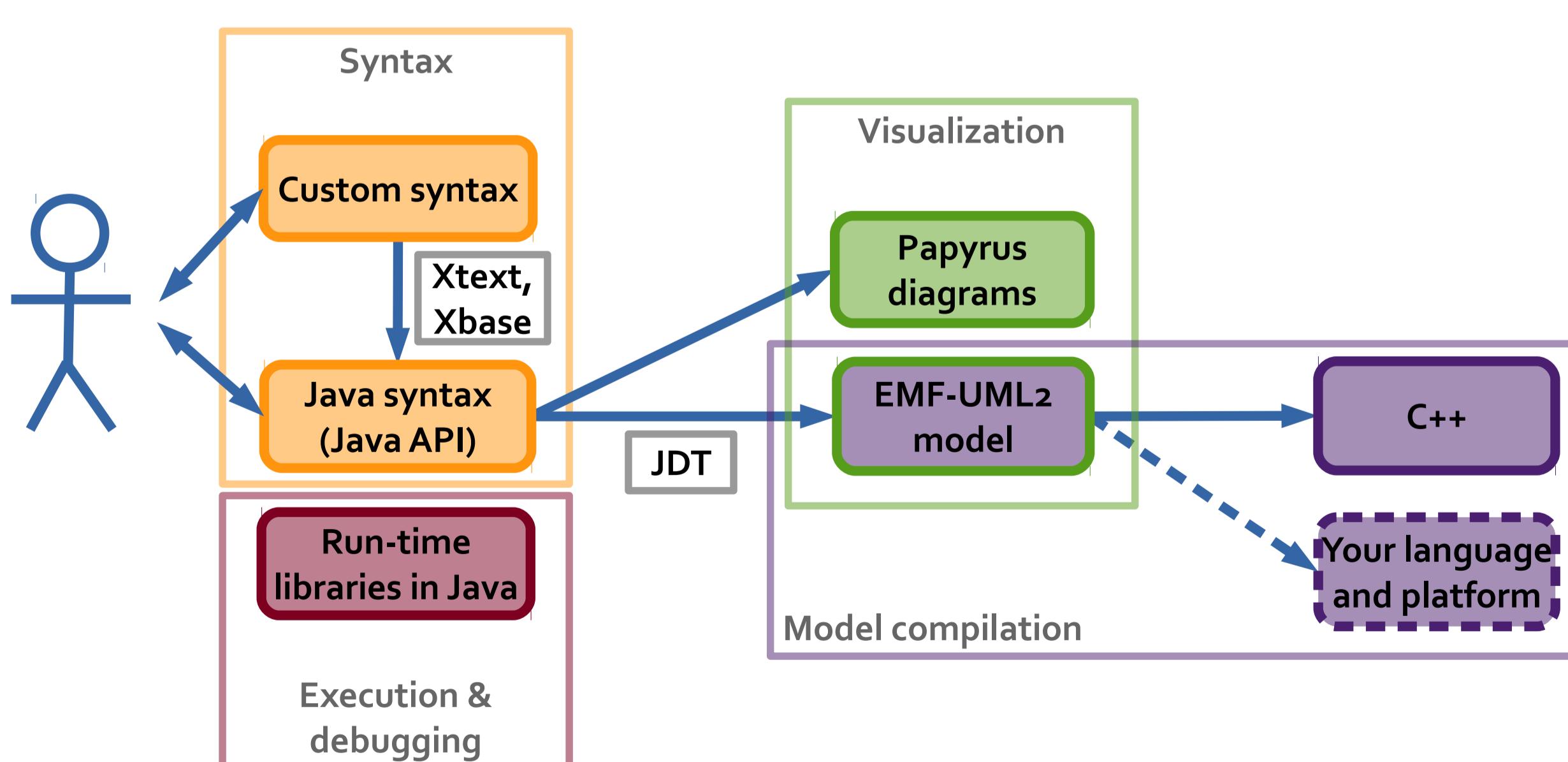
The usual debugging features (breakpoints, pause, resume, step, etc.)  
are available for model debugging.



State machine  
diagrams can  
be animated.

### Translatable

Experimental model compiler for C++ is available,  
support for other languages and platforms is possible.



### UML

UML is a standard, well-known language  
with all necessary elements for executable modeling.

What is the supported UML subset?

- **Class modeling:** classes, attributes, methods, binary associations, compositions, generalization
- **State modeling:** simple states, hierarchical states, guards, choice nodes, entry, exit and effect activities
- **Component modeling:** interfaces, ports, assembly and delegation connectors
- **Diagrams:**
  - Class and state machine diagrams
  - In preparation: sequence diagrams, composite structure diagrams



Web: [txtuml.inf.elte.hu](http://txtuml.inf.elte.hu)

GitHub: [github.com/ELTE-Soft/txtUML](https://github.com/ELTE-Soft/txtUML)

Mail: [txtuml@inf.elte.hu](mailto:txtuml@inf.elte.hu)

{txtUML}  
textual, executable, translatable