# Direct Model-Checking of SysML Models

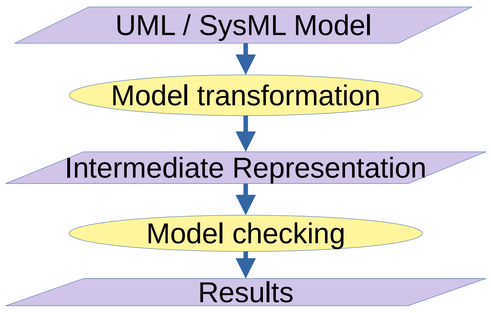
Alessandro TEMPIA CALVINO[1,2], Ludovic APVRILLE[1]

1. LTCI, Telecom Paris, Institut Polytechnique de Paris, Sophia-Antipolis, France
2. Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland

## Model-checking from SysML

### Using an intermediate representation

- Model is first transformed into a formal specification
- Formal specification is then fed into an external tool
- Results are backtraced to the model
- UML-to-PN, UML-to-LOTOS, SysML-to-UPPAAL, ...

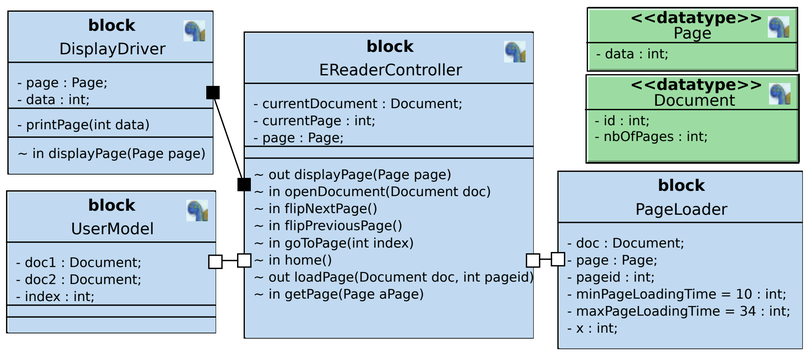



### Our idea

- Model-checking from SysML models
- Avoid transformations
- Make the backtracing much easier to perform
- Directly integrated in the SysML toolkit TTool

## Contribution: Model-checking Approach

### Case study: An E-book Reader

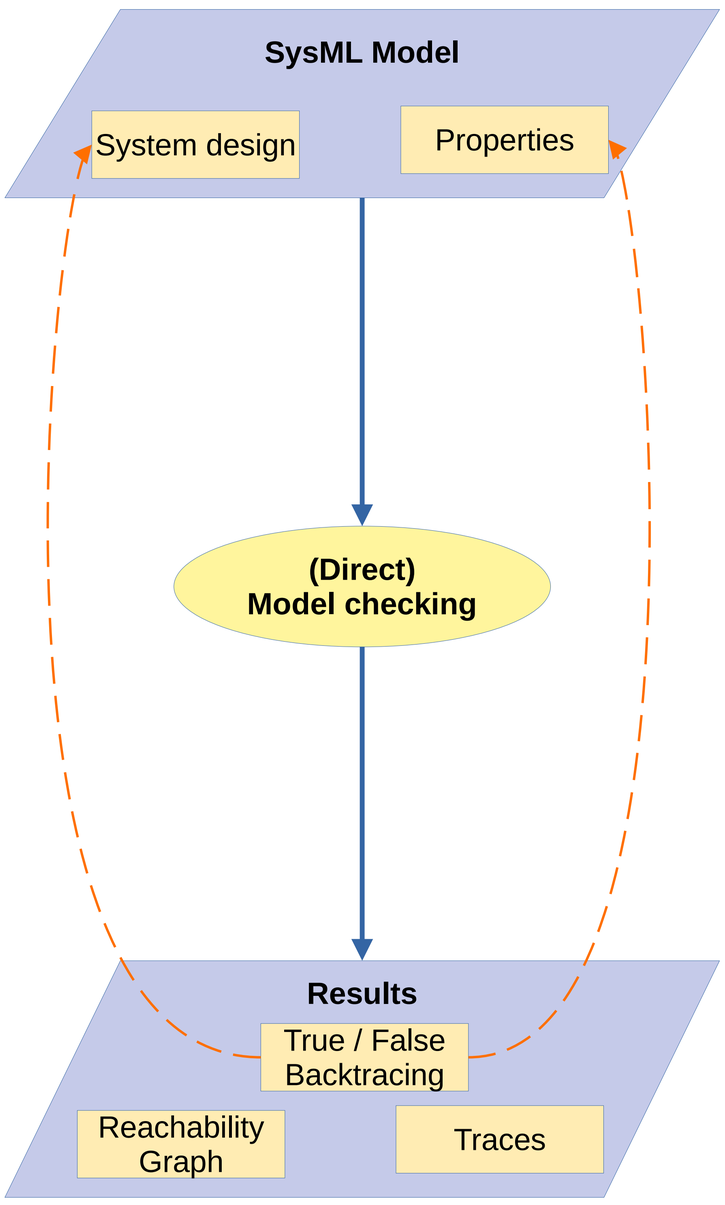



### Main algorithm

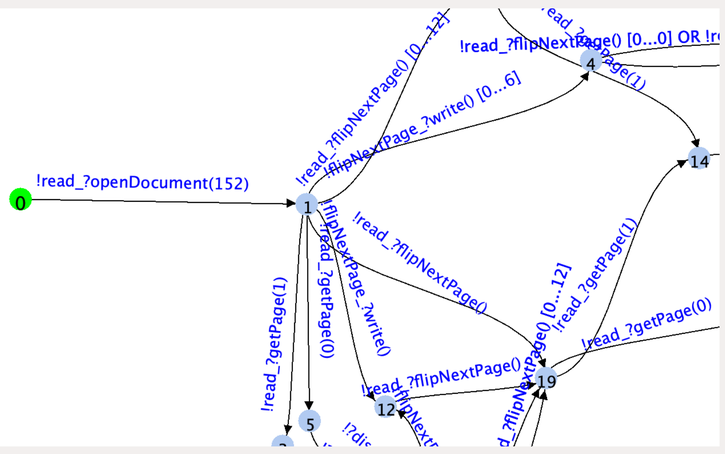
1: $S_0 = \{s_0^0, s_0^1, \ldots, s_0^N\}$ ▷ Initial r-state
2: *STATES*=$\{S_0\}$, *PENDING*=$\{S_0\}$
3: **while** *PENDING*$\neq \{\}$ **do**
4: Extract a state $S$ from the *PENDING* queue
5: $\mathcal{T}$ = executable transitions from r-state $S$
6: **for** each transition $t_i$ in $\mathcal{T}$ **do**
7: Execute $t_i$ obtaining a r-state $P$
8: Evaluate properties $(S, t_i, P)$
9: **if** $P \notin$ *STATES* **then** ▷ New r-state
10: Add $P$ to *STATES*
11: Append $P$ to *PENDING*
12: **end if**
13: Add a new edge $S \rightarrow P$
14: **end for**
15: **if** $\mathcal{T} = \{\}$ **then** ▷ Deadlock
16: Evaluate properties on deadlock
17: **end if**
18: **end while**

### Algorithm optimizations

- Multi-threaded BFS or DFS exploration
- States and transitions merging techniques
- Compact state encoding and hashing
- Division in time domains




### Reachability Graph




### Properties

```
Safety Pragmas
T UserModel.FlipPage --> DisplayDriver.NewPage
F A[] PageLoader.x<12
T A[] PageLoader.x<13
T E[] PageLoader.x==12||PageLoader.x==0
T A<> EReaderController.currentPage==0&&DisplayDriver.NewPage
F E<> PageLoader.x == 13
```

- Reachability of states, Liveness and safety, Deadlock freedom
- CTL formulae
  - *AG p*
  - *AF p*
  - *EG p*
  - *EF p*
  - *AG ( p ⇒ AF q )*

### Handling Properties

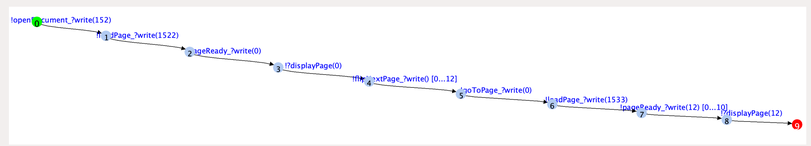
- On-the-fly during the generation of the reachability graph (RG)
- Reachable states are reported during the RG creation
- Deadlock when a r-state has no executable transition
- On-the-fly cycle detection to prove properties

### Backtracing

```
Safety Pragmas
✓ T UserModel.FlipPage --> DisplayDriver.NewPage
✓ F A[] PageLoader.x<12
✓ T A[] PageLoader.x<13
✓ T E[] PageLoader.x==12||PageLoader.x==0
✗ T A<> EReaderController.currentPage==0&&DisplayDriver.NewPage
✓ F E<> PageLoader.x == 13
```

### Traces

- Illustrate why a property is or is not satisfied




### Performance Evaluation

| *Model* | *Property* | Time TTool (ms) | Time UPPAAL (ms) |
|---|---|---|---|
| *ebook* | *RG* | 3832 | / |
| *ebook* | *D* | 14 | 755 |
| *ebook* | *LeadsTo* | 5659 | 2235 |
| *ebook* | *A[]x<12 (BFS)* | 10 | 287 |
| *ebook* | *A[]x<13 (BFS)* | 3478 | 1303 |
| *ebook* | *E[] (BFS)* | 14 | 282 |
| *ebook* | *A<>(DFS)* | 29 | 279 |
| *ebook* | *E<>(BFS)* | 3604 | 1292 |
| *ebook* | *R, L, D, CTL* | 17529 | 12068 |

*(More comparisons are given in the paper)*

### Conclusion and Future work

- New model-checker already available in TTool
- Performance similar to the one of UPPAAL
- Extension to other profiles supported by TTool (e.g. DIPLODOCUS)
- New optimization techniques
- Integration within other frameworks


*This work is part of the AQUAS project which is funded by ECSEL JU under grant agreement No 737475*


Contacts: alessandro.tempiacalvino@epfl.ch ludovic.apvrille@telecom-paris.fr