Model-checking with the TimeLine formalism

Andrea Zaccara

Andrea.Zaccara@student.uantwerpen.be

University of Antwerp

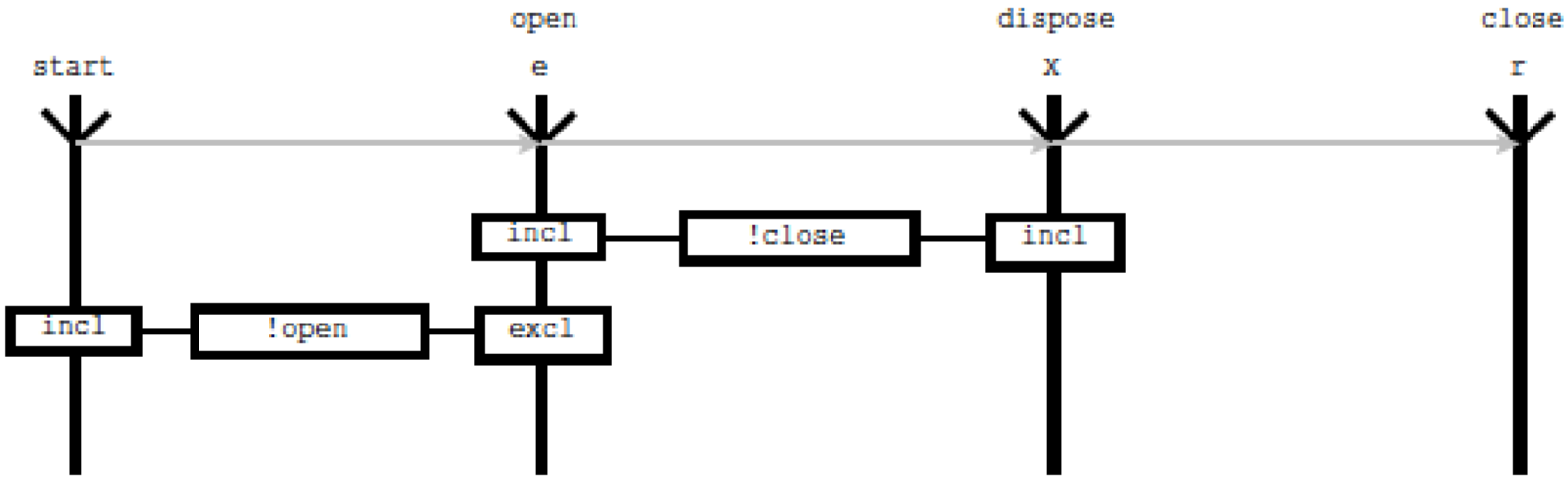
The problem

- Model checking is an useful tool for the verification of software requirements
- The specification definition use a cumbersome textual notation
- The definition of temporal logic requirements is limited to verification experts, hindering its adoption in software development

The TimeLine formalism

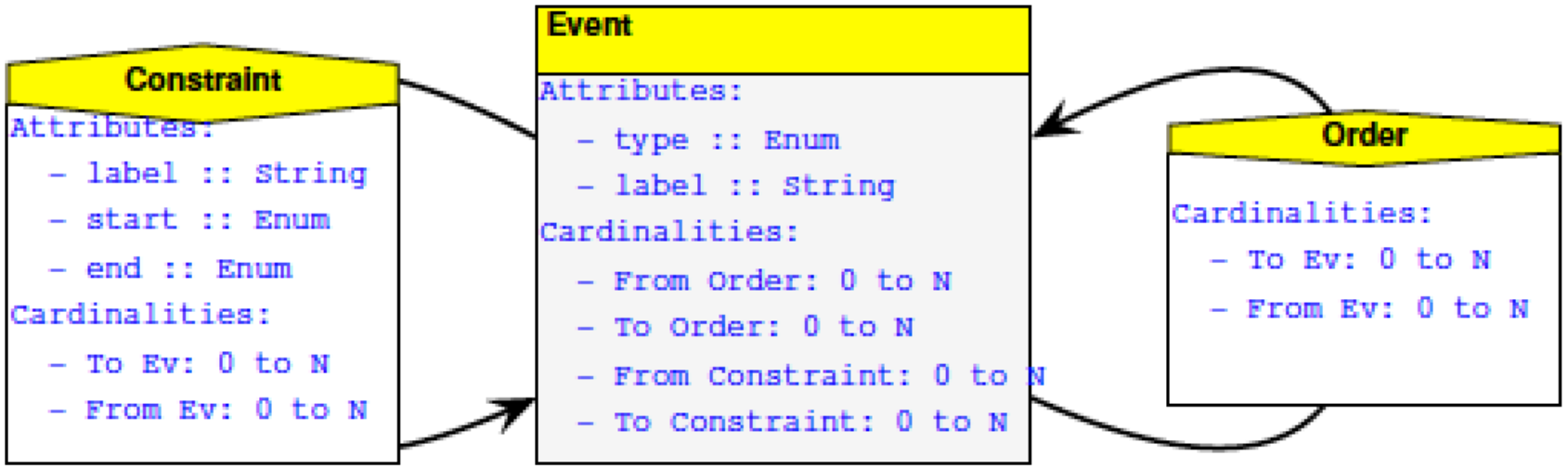
Defines a sequence of events, categorized by a type:

- Regular (e), used to create a context for following events
- Required (r), this event must occur if previous events in the time line have occurred
- Fail (X), this event must not occur if previous events in the time line have occurred



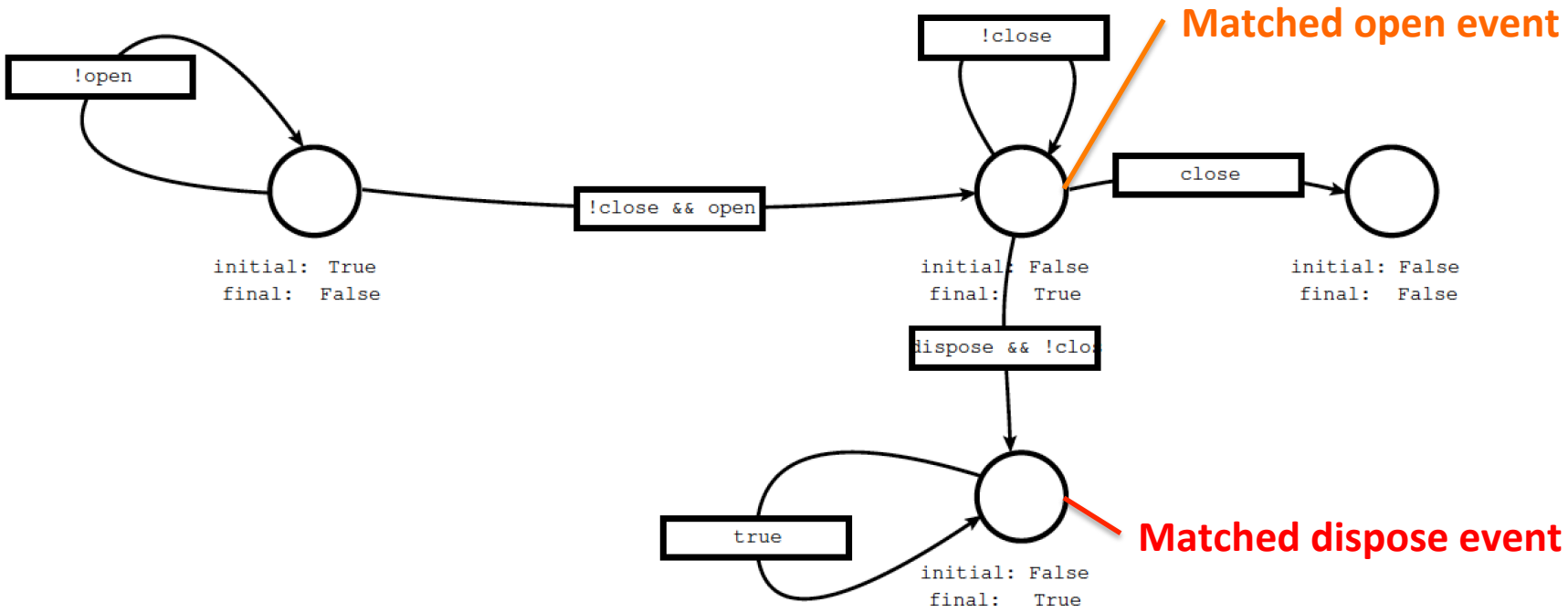
Abstract syntax

- Defined as an ordered sequence of fully connected events
- Additional single *start* and *end* events
- Definition of constraint in-between events



Transformation to Automaton

- Model transformations generate a finite state automaton from TimeLine specification
- This has used for runtime monitoring and model-checking



E. Bodden, H. Vangheluwe. Transforming timeline specifications into automata for runtime monitoring, in: Applications of Graph Transformations with Industrial Relevance, Springer, 2008, pp. 249-264.

Project scope

- Definition of the TimeLine meta-model in AToMP
- Model transformations to automaton and code-generation for regular expression recognition
- Application to the verification of requirements for the communication protocol of a client-server chat system

Questions?