Model-driven rapid prototyping with Umple

Andrew Forward, Omar Badreddin, Timothy C. Lethbridge*, and Julian Solano

School of Electrical Engineering and Computer Science, University of Ottawa,

Presenter: Maryam Davari

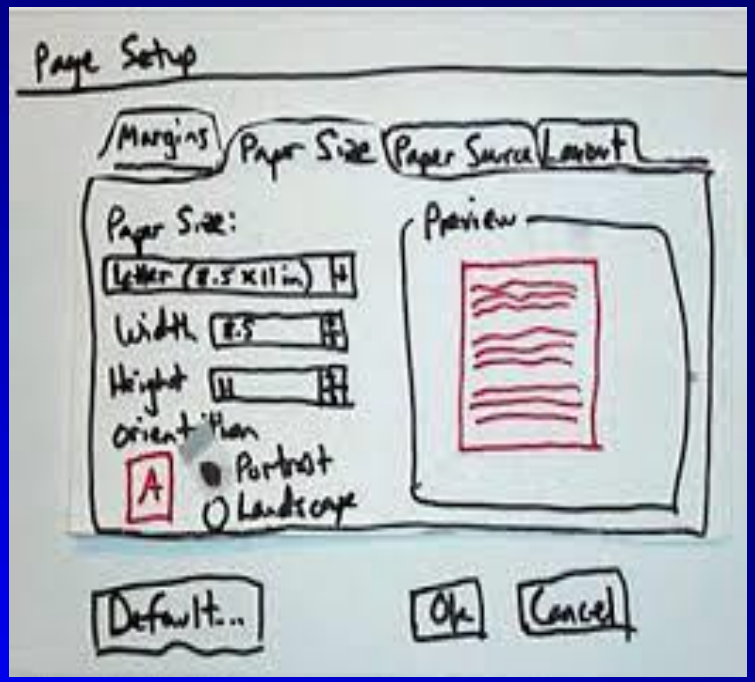
Outline

- Prototype
- Objective of the paper
- Model-driven software development (MDSD)
- MDSD challenges for prototyping
- Umple
- UmpleOnline
- Class Diagram in Umple
- The Umple user-interface prototype generator

Outline (cont.)

- The Umple modeling and prototyping approach
- State Machine in Umple
- Mixin in Umple
- Umple's architecture
- Related work for generation of prototypes and user interfaces
- Conclusion

Prototype


Page Setup
Margins
Paper Size
Paper Source
Layout
Paper Size:
Letter (8.5 x 11 in)
Width
8.5
Height
11
Orientation
Portrait
Landscape
Preview
Default...
Ok
Cancel


VMS Video Downloads
TV Time Machine
TV Discover Tools
2000's
1990's
1980's
1970's
1960's
1950's
Most Popular Shows from the 90's
Select a Year from the 90's:

Prototype (cont.)

- Has an indispensable role in software industry.
- Creates a manifestation in the simplest form.
- Filters qualities without distorting the understanding.

Objective of the paper

- How to prototype the essential aspects.
- focus on
  - application-domain data (UML class diagram).
  - behavior (state machine).
- Filters details such as algorithms and mechanisms.

Model-driven software development

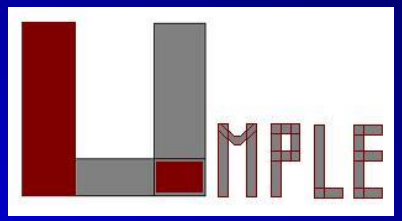
- Is an alternative to round-trip engineering.
- Uses languages such as UML.
- Generates executable system.
- Is inherently possible to generate prototypes from models.

MDSD challenges for prototyping

- Models are not intended for modeling the user interface.
- Combining generated codes from models with other codes is not easy.
- There is a need to generate tangible artifact.
- Software modelers do not understand the consequences of modeling decision.
- Limitations in approaches to prototype generation from UML models.

Umple

- Stands for “UML Programming Language”.

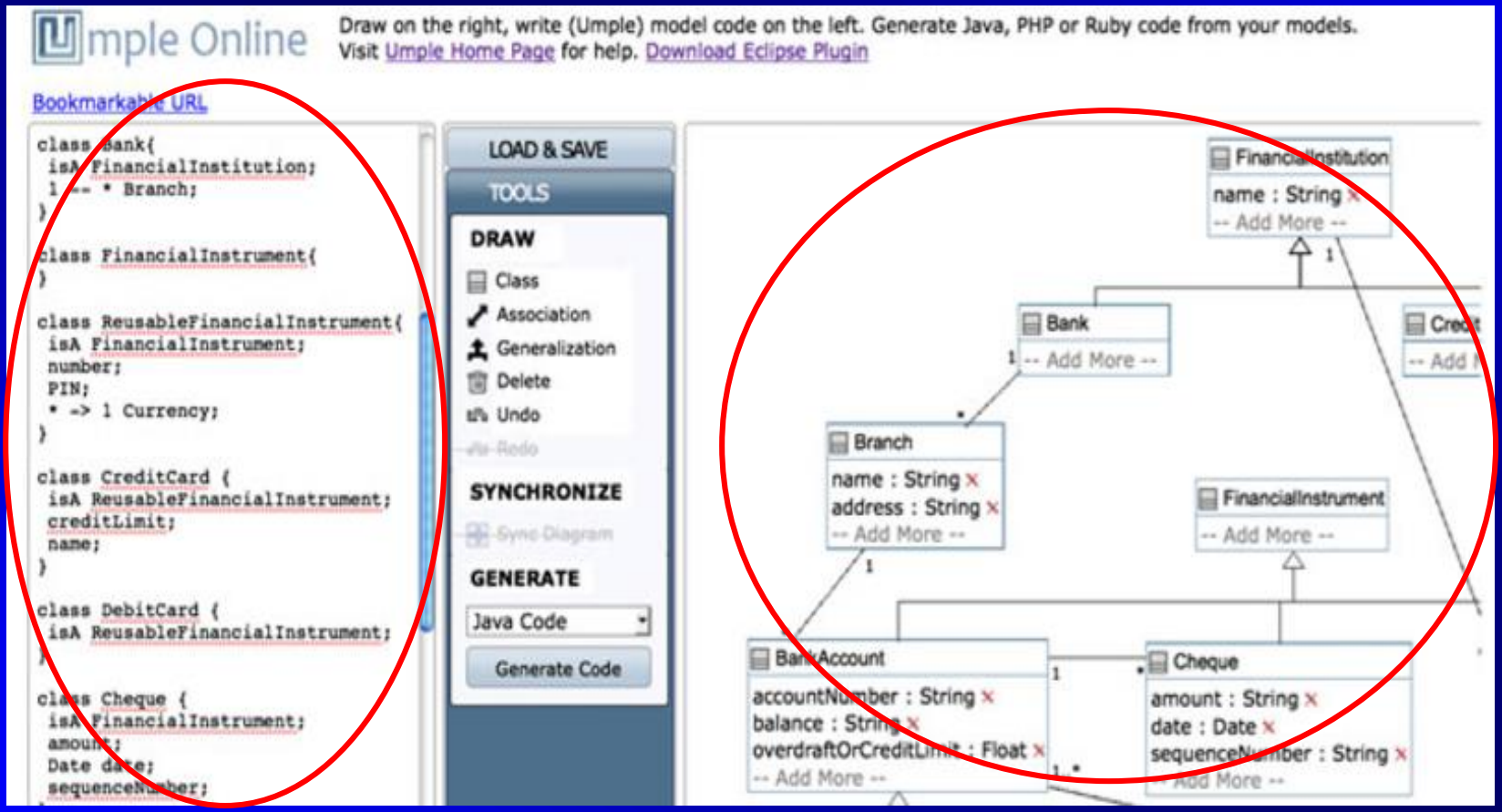


=

+

- Can be embedded in several programming languages.

UmpleOnline



The UmpleOnline web application showing a UML mode being edited textually and graphically

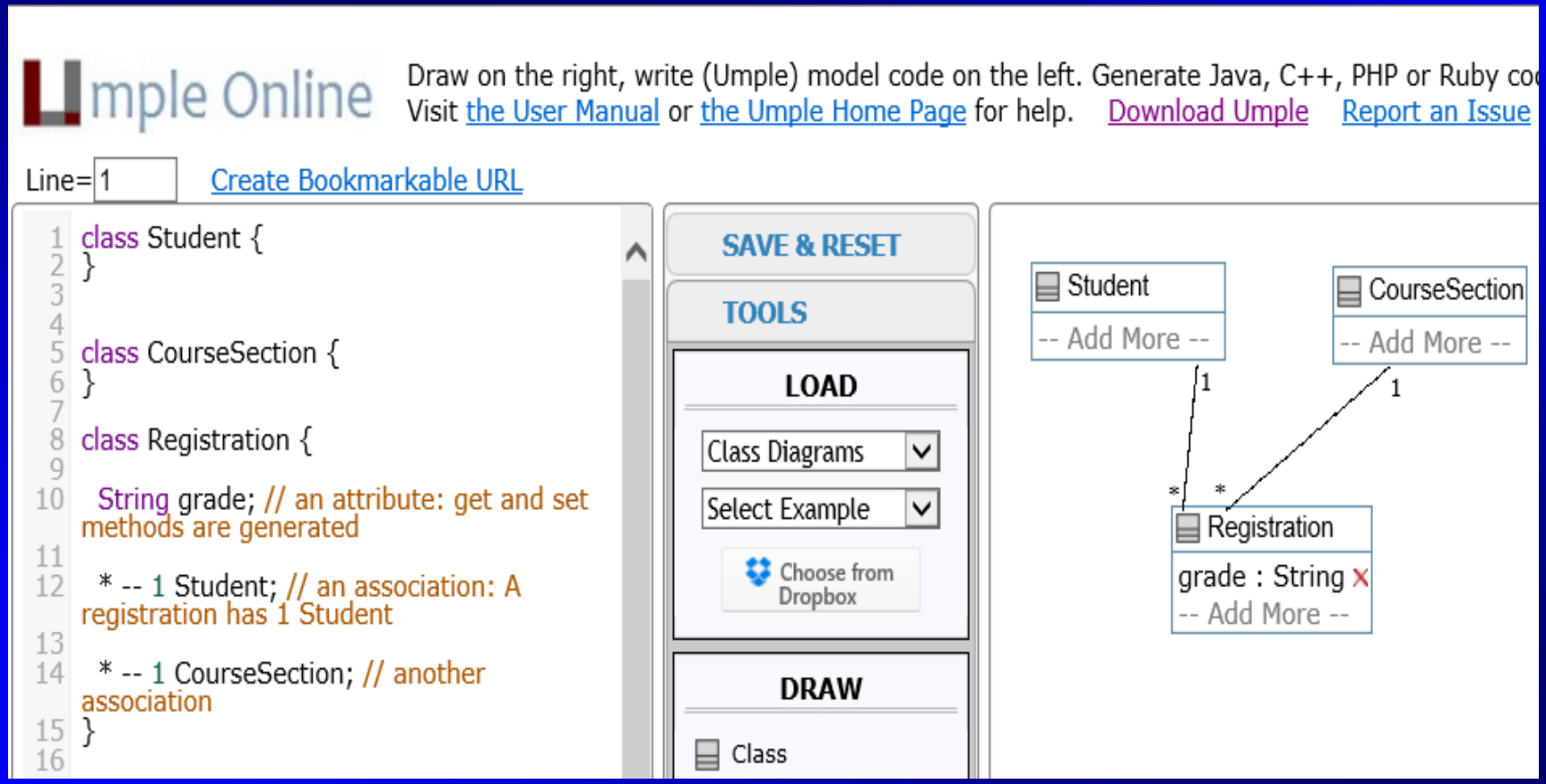
Umple (cont.)

- Supports either textually and graphically.
- Textual editors supports:
  - Syntax-Driven editing
  - Searching
  - Auto indentation
  - Mixins
- Graphical Models Support:
  - Better communication & collaboration.
  - Relationships are clearer.

Umple (cont.)

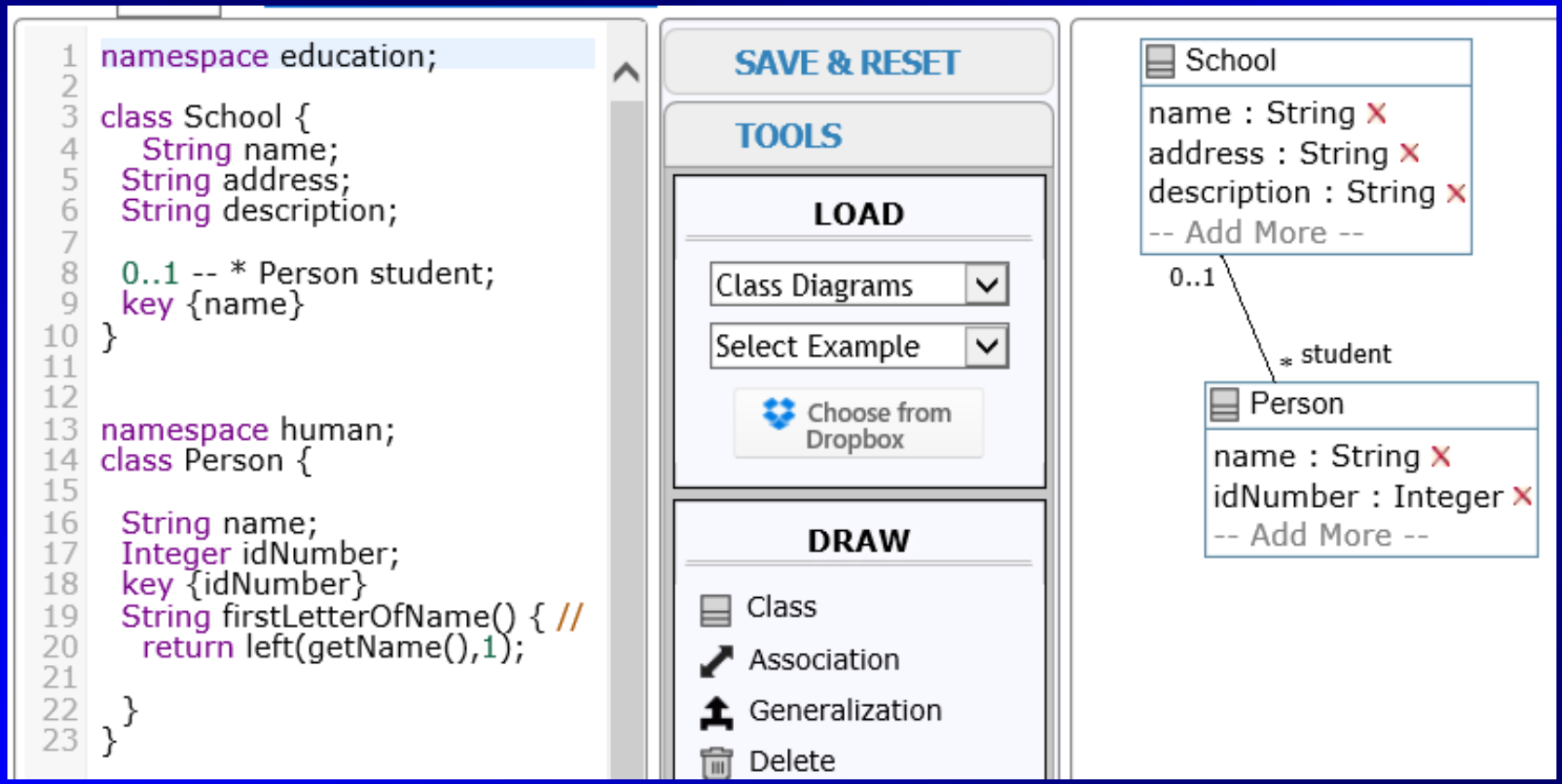
- Provides a tool to edit UML class diagram in the graphical form.
- Provides a set of textual notations for UML modeling abstractions such as classes, associations, states, transitions.
- Allows end users to quickly create class diagram, state machine and Mixins.

Class Diagram in Umple



A class diagram corresponding to Umple code.

Example of Umple



Two classes in two different namespaces with corresponding diagrams

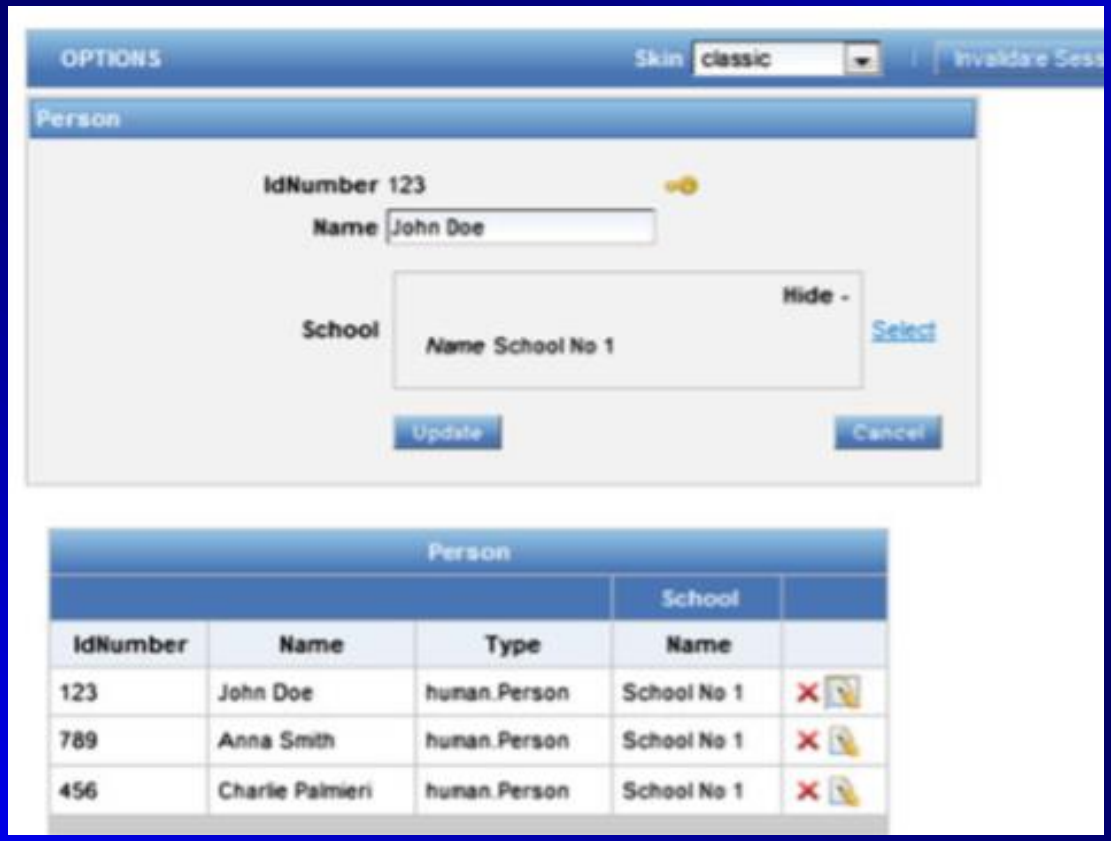
The Umple user-interface prototype generator

- At any point in the modeling or development process, users can quickly generate a prototype.
- Users can create instance of classes and links of associations.
- Semantic rules are respected at run-time.
- User can follow links.
- Users can view and update object's attributes.
- Users can change predefined theme.

The Umple modeling and prototyping approach (cont.)

Prototype interface for creating, editing and deleting instances of the Person class

The Umple modeling and prototyping approach (cont.)



An instance of Person with a linked school.

State Machine in Umple

- Umple supports the specification of state machine.
- Adheres to UML semantic except one condition.
- Provides compact representation of behavior, and is ideal for rapid prototyping.
- Is optional feature of Umple.

State Machine in Umple

```
1  class trafficLightSystem
2  {
3   carTraffic{
4
5     Red{
6       entry /  {goingRed();}
7       after(redTimer)[!emergency] -> Yellow;
8
9       emergencyNotice -> AllRed;
10    }
11    Yellow{
12    }
13  }
14
15  pedestrainTraffic{
16
17    DontWalk{
18
19      goingRed[!emergency] -> Walk;
20
21      emergencyNotice -> DonotWalk;}
22  }
23 }
```

SAVE & RESET

TOOLS

LOAD

Class Diagrams

Select Example

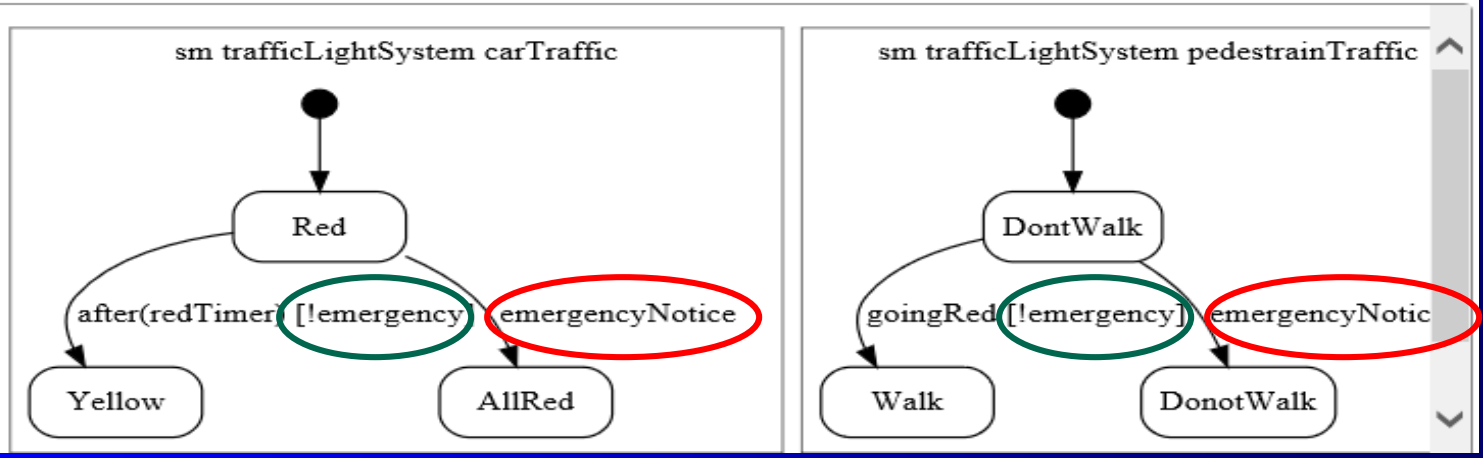
Choose from Dropbox

DRAW

Class

Association

OPTIONS

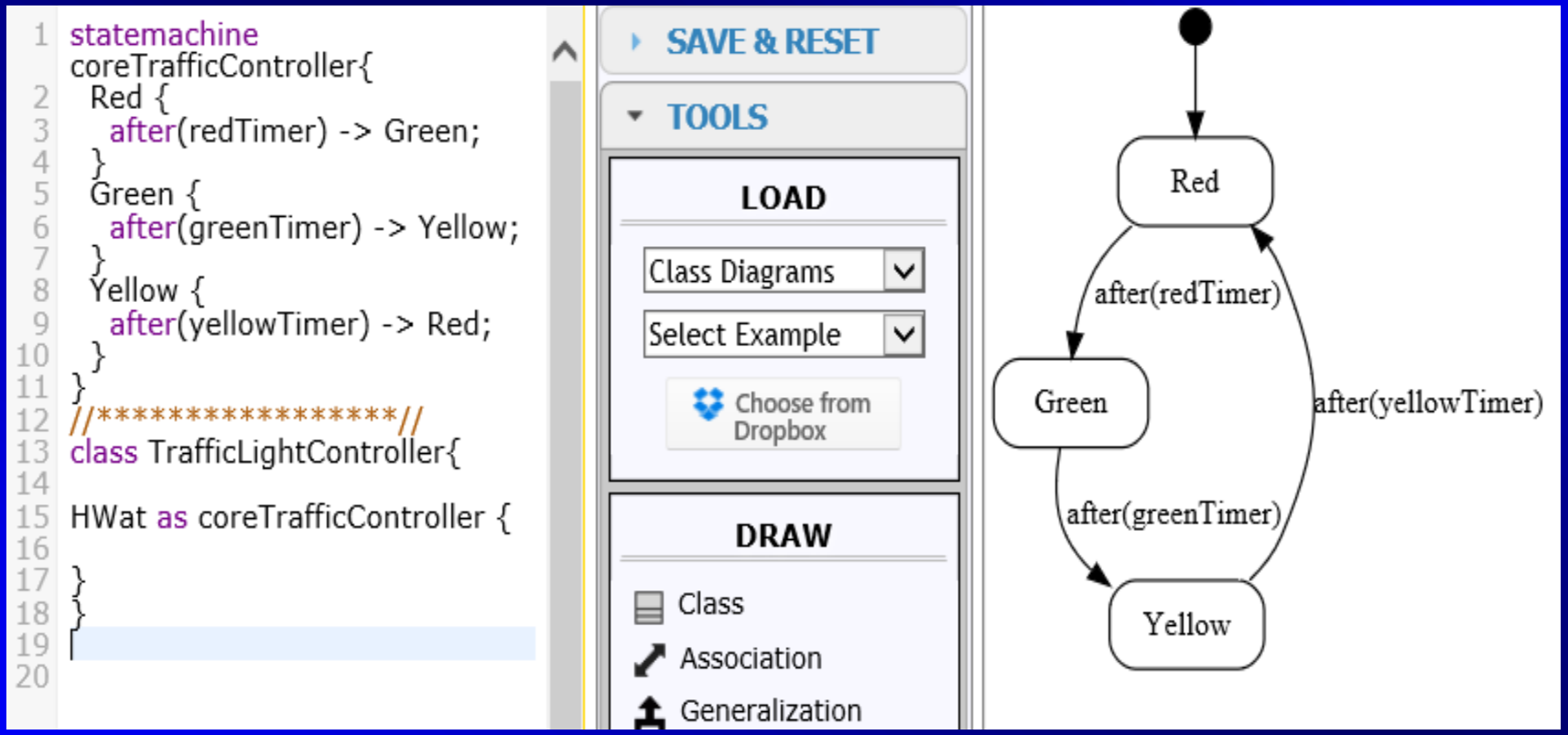


Example of state machine

Mixin in Umple

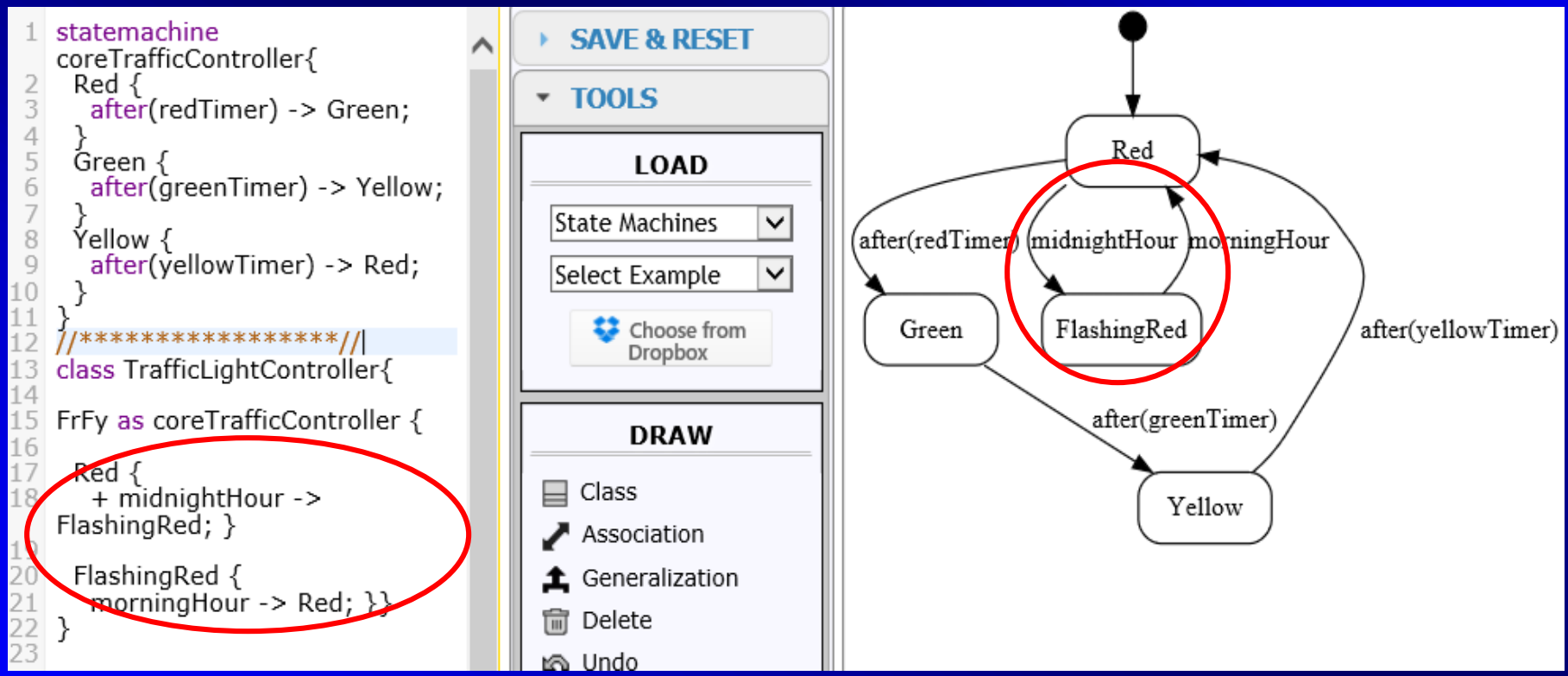
- Enables to inject developed codes into a set of classes.
- Useful for prototype development.
- Enhances reusability of components.
- Can be applied to state machines.

Mixin in Umple (cont.)



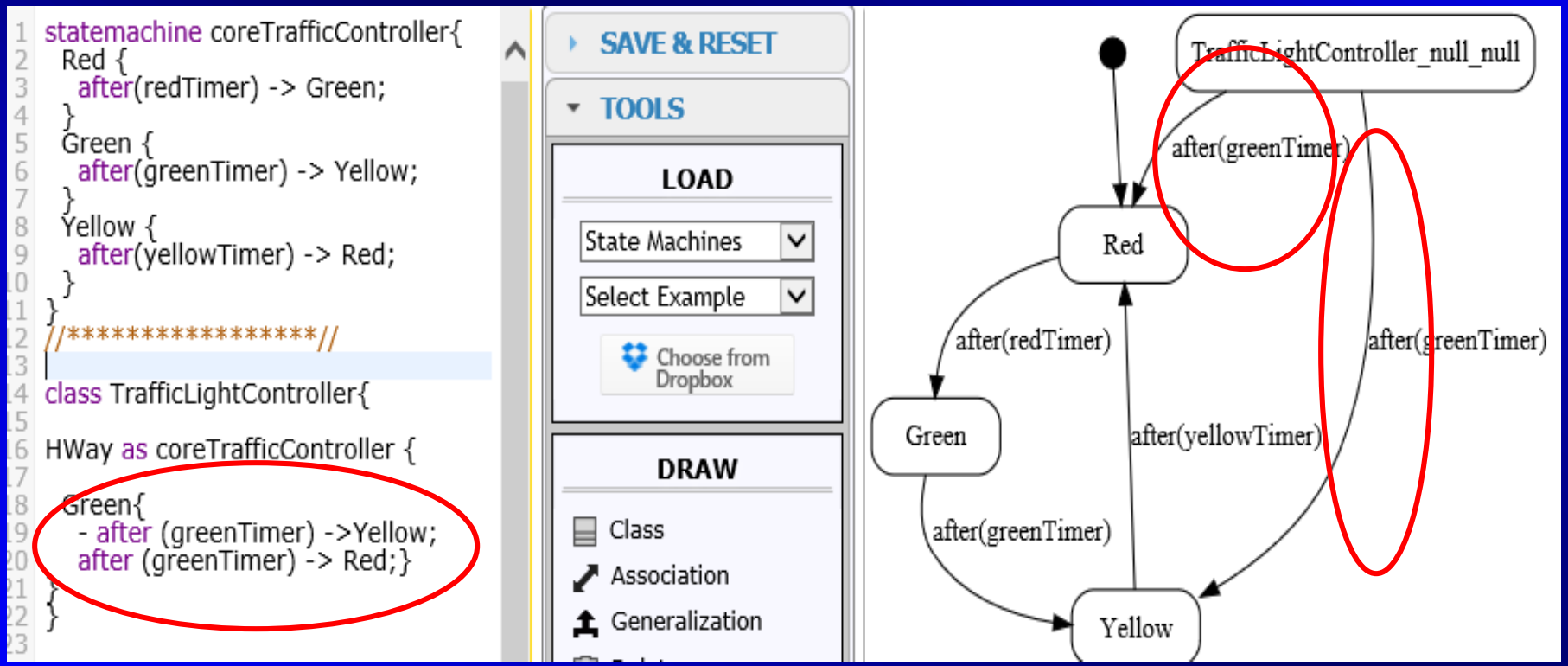
An example of Mixin in state machine

Mixin in Umple (cont.)



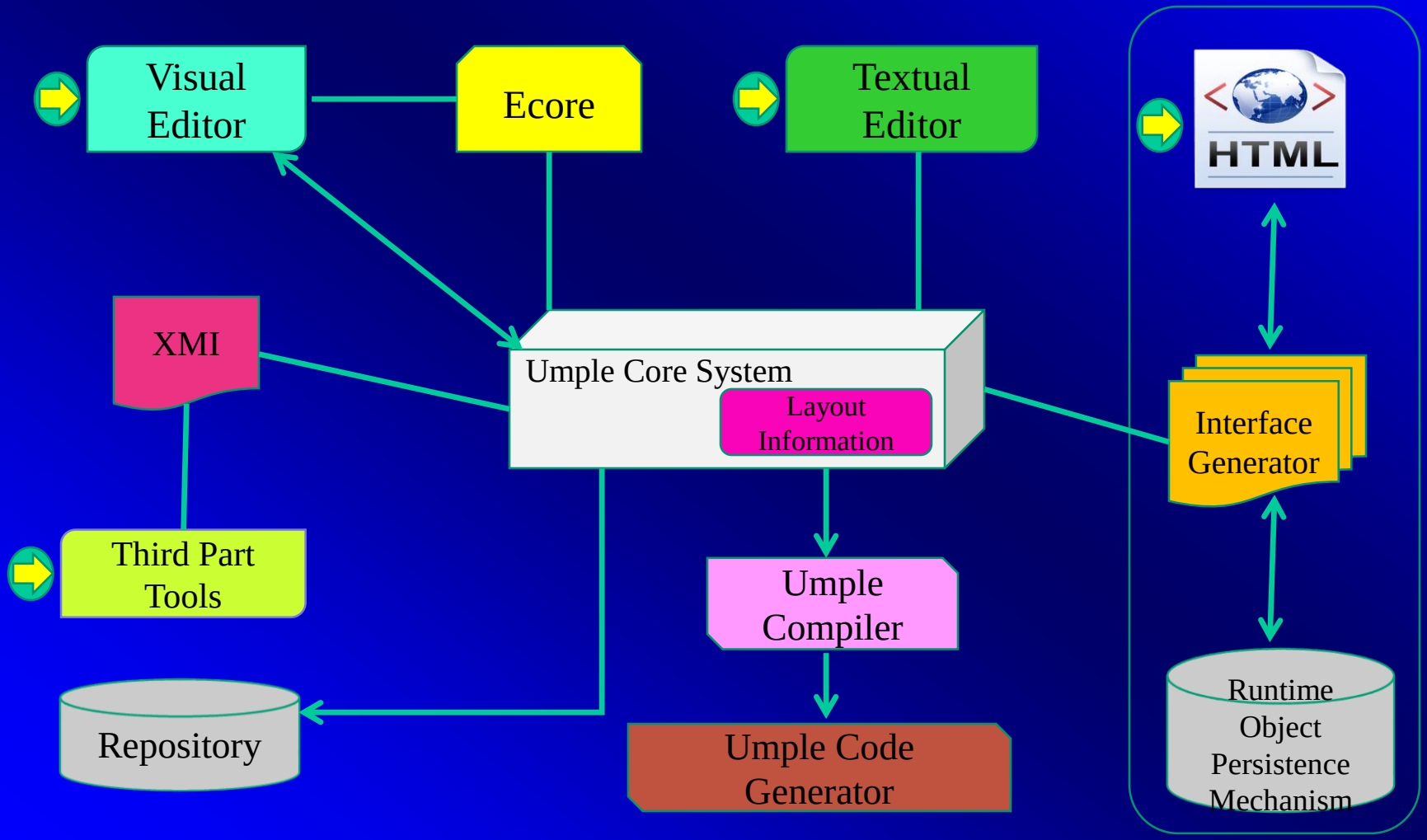
Adding additional states that are not part of the basic state machine

Mixin in Umple (cont.)



Removing or replacing a transition

Umple's architecture



Related work for generation of prototypes and user interfaces

- Generate UI from:
  - State machine models
  - behavior of the system without considering data.
  - Data structure
  - Use case

Conclusion (cont.)

- Umple provides:
  - Accurate reflection of system components and data.
  - Accurate reflection of current system design directions.
  - Quick and cheap generation of a prototype.
  - Maximized potential for reuse and composition.
  - Support for different levels of abstraction.
  - Support for incremental development.

Thank you