

# The Eclipse Modeling Framework

## Modeling Metamodels

Christoph Daniel Schulze



Real-Time Systems and Embedded Systems Group  
Department of Computer Science  
Faculty of Engineering  
Christian-Albrechts-Universität zu Kiel

October 29th, 2012

# Outline

- 1 An Example: Regular Expressions
- 2 The Eclipse Modeling Framework

# Outline

① An Example: Regular Expressions

② The Eclipse Modeling Framework

# Definition

## Regular Expressions over Given Alphabet $\Sigma$

# Definition

## Regular Expressions over Given Alphabet $\Sigma$

- Terminals:

# Definition

## Regular Expressions over Given Alphabet $\Sigma$

- Terminals:
  - $\varepsilon$ : The empty set / empty String

# Definition

## Regular Expressions over Given Alphabet $\Sigma$

- Terminals:
  - $\varepsilon$ : The empty set / empty String
  - $\alpha \in \Sigma$ : Literal character

# Definition

## Regular Expressions over Given Alphabet $\Sigma$

- Terminals:
  - $\varepsilon$ : The empty set / empty String
  - $\alpha \in \Sigma$ : Literal character
  
- Compound expressions:  
Let  $R$  and  $S$  be expressions; then:



# Definition

## Regular Expressions over Given Alphabet $\Sigma$

- Terminals:
  - $\varepsilon$ : The empty set / empty String
  - $\alpha \in \Sigma$ : Literal character
  
- Compound expressions:  
Let  $R$  and  $S$  be expressions; then:
  - *Concatenation*:  $RS$  matches  $\{rs : r \in R \wedge s \in S\}$

# Definition

## Regular Expressions over Given Alphabet $\Sigma$

- Terminals:
  - $\varepsilon$ : The empty set / empty String
  - $\alpha \in \Sigma$ : Literal character
  
- Compound expressions:  
Let  $R$  and  $S$  be expressions; then:
  - *Concatenation*:  $RS$  matches  $\{rs : r \in R \wedge s \in S\}$
  - *Alternation*:  $R|S$  matches either  $R$  or  $S$

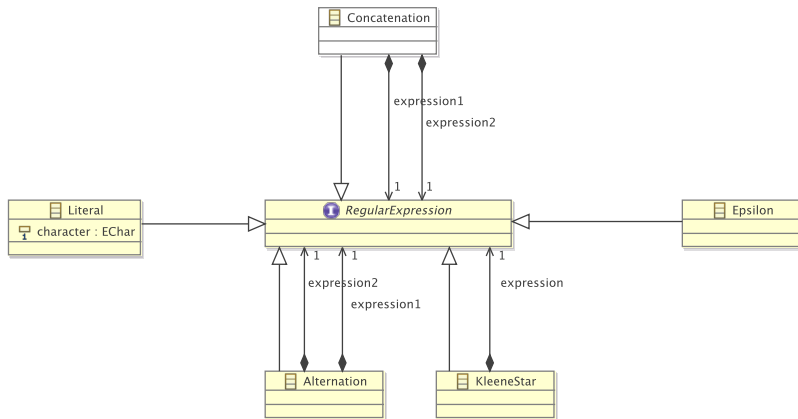
# Definition

## Regular Expressions over Given Alphabet $\Sigma$

- Terminals:
  - $\varepsilon$ : The empty set / empty String
  - $\alpha \in \Sigma$ : Literal character
  
- Compound expressions:  
Let  $R$  and  $S$  be expressions; then:
  - *Concatenation*:  $RS$  matches  $\{rs : r \in R \wedge s \in S\}$
  - *Alternation*:  $R|S$  matches either  $R$  or  $S$
  - *Kleene star*:  $R^*$  matches any finite repetition of elements in  $R$

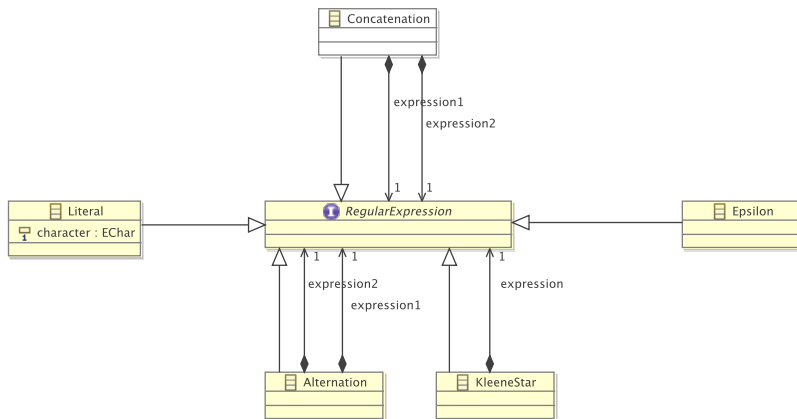
# Modeling Regular Expressions with EMF

## One Possible Solution



# Modeling Regular Expressions with EMF

## One Possible Solution



This is not the only possible solution!

# Outline

① An Example: Regular Expressions

② The Eclipse Modeling Framework

# The EMF Workflow

A Recipe for Modeling with EMF

# The EMF Workflow

## A Recipe for Modeling with EMF

- 1 Find something to model



# The EMF Workflow

## A Recipe for Modeling with EMF

- 1 Find something to model
- 2 Think about how it is structured

# The EMF Workflow

## A Recipe for Modeling with EMF

- 1 Find something to model
- 2 Think about how it is structured
- 3 Design a grammar accordingly (*Metamodel*)

# The EMF Workflow

## A Recipe for Modeling with EMF

- 1 Find something to model
- 2 Think about how it is structured
- 3 Design a grammar accordingly (*Metamodel*)
- 4 Generate a generator model for the metamodel

# The EMF Workflow

## A Recipe for Modeling with EMF

- 1 Find something to model
- 2 Think about how it is structured
- 3 Design a grammar accordingly (*Metamodel*)
- 4 Generate a generator model for the metamodel
- 5 Generate code and be happy

# Some Details

## Pitfalls in Metamodeling

# Some Details

## Pitfalls in Metamodeling

- Containment references

# Some Details

## Pitfalls in Metamodeling

- Containment references
- Opposites

# Some Details

## Pitfalls in Metamodeling

- Containment references
- Opposites
- Multiplicities