



# Politechnika Wrocławska

## **Model-to-Text (M2T)** **Generowanie zapisu równań** **matematycznych z modelu EMF**



Aleksandra Chrustek, Julia Ujma, Gabrysia Czernecka

# Cel projektu

Celem projektu jest automatyczne generowanie równań w formie tekstowej na podstawie abstrakcyjnego modelu danych.

Projekt pokazuje, że model może być punktem wyjścia do tworzenia rzeczywistego tekstu, a nie tylko rysunkiem. Dzięki temu można budować bardziej zautomatyzowane, mniej podatne na błędy systemy.

Kroki realizacji projektu:

1. Utworzenie metamodelu wraz z relacjami między wyrażeniami i operacjami.
2. Stworzenie instancji modelu (.xmi).
3. Zaprojektowanie szablonu .mtl w Acceleo.
4. Uruchomienie generacji.
5. Testy i poprawki.

# Metamodel równań

Metamodel reprezentuje drzewo wyrażenia. Zawiera abstrakcyjną klasę ogólną **Expression** oraz klasy dziedziczące po niej:

- Number
- Variable
- Addition
- Multiplication
- Division
- Sqrt
- Constant
- Power
- Modulo
- Factorial
- Absolute
- Sum
- Integral
- Product
- FunctionCall
- Parenthesized
- MatrixExpr
- MatrixContent
- MatrixRow

Każda operacja ma swoje atrybuty i odniesienia do innych Expression.



# Metamodel równań - Ecore

- ▼ math
  - Expression
  - ▼ Number -> Expression
    - value : EDouble
  - ▼ Variable -> Expression
    - name : EString
  - ▼ Addition -> Expression
    - left : Expression
    - right : Expression
  - ▼ Multiplication
    - left : Expression
    - right : Expression
  - ▼ Division
    - numerator : Expression
    - denominator : Expression
  - ▼ Sqrt
    - radicand : Expression
  - ▼ Constant -> Expression
    - name : EString
  - ▼ Power -> Expression
    - base : Expression
    - exponent : Expression
- ▼ Modulo -> Expression
  - left : Expression
  - right : Expression
- ▼ Factorial -> Expression
  - value : Expression
- ▼ Absolute -> Expression
  - value : Expression
- ▼ Sum -> Expression
  - lower : Expression
  - upper : Expression
  - body : Expression
- ▼ Integral -> Expression
  - lower : Expression
  - upper : Expression
  - body : Expression
- ▼ Product -> Expression
  - lower : Expression
  - upper : Expression
- ▼ FunctionCall -> Expression
  - func : EString
  - argument : Expression
- ▼ Parenthesized -> Expression
  - expr : Expression
- ▼ MatrixExpr -> Expression
  - content : MatrixContent
- ▼ MatrixContent
  - rows : MatrixRow
- ▼ MatrixRow
  - elements : Expression



# Model wyrażenia

Kolejnym krokiem jest stworzenie przykładowego **modelu wyrażenia** (instancji) .xmi.



# Szablon Model-to-Text Language

Następnie należy stworzyć **szablon .mtl**. Plik .mtl to szablon Acceleo (Model-to-Text Language), który pozwala generować tekst z modeli EMF (np. .xmi).

Szablon .mtl działa podobnie jak szablon HTML pod względem logiki - posiada instrukcje warunkowe, pętle i wywołania szablonów.

Uruchomienie pliku szablonu wraz z modelem wyrażenia .xmi wygeneruje plik docelowy zawierający **zapis matematyczny w formie tekstowej**.



# Politechnika Wrocławska

**Dziękujemy za uwagę**



Aleksandra Chrustek, Julia Ujma, Gabrysia Czernecka