UVLParser: Extending UVL with Language Levels and Conversion Strategies

SPLC'23 | Chico Sundermann, Stefan Vill, Thomas Thüm, **Kevin Feichtinger**, Prankur Agarwal, Rick Rabiser, José A. Galindo, David Benavides | 31.08.2023

universität ulm

JOHANNES KEPLER UNIVERSITÄT LINZ

University of Seville

Universal Variability Language

- Community Effort within MODEVAR initiative
- Textual format for variability models
- Simplify exchange

```
features
    Pizza
        mandatory
            Dough {Calories 10, Type 'Wheat'}
            Cheese {Calories 5}
            Sauce
                alternative
                    Tomato {Calories 2}
                    Pesto {Calories 4}
        optional
            Mushrooms {Calories 1}
            Ham {Calories 7}
            Pineapple {Calories 2}
            "Greetings on box"

constraints
    Pineapple => Ham
```

Universal Variability Language

- Community Effort within MODEVAR initiative
- Textual format for variability models
- Simplify exchange
- Simple core language
  - Boolean constraints & features

```
features
    Pizza
        mandatory
            Dough {Calories 10, Type 'Wheat'}
            Cheese {Calories 5}
            Sauce
                alternative
                    Tomato {Calories 2}
                    Pesto {Calories 4}
        optional
            Mushrooms {Calories 1}
            Ham {Calories 7}
            Pineapple {Calories 2}
            "Greetings on box"

constraints
    Pineapple => Ham
```

Universal Variability Language

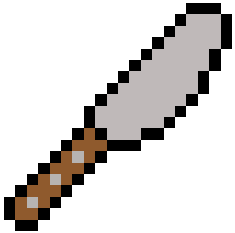
- Community Effort within MODEVAR initiative
- Textual format for variability models
- Simplify exchange
- Simple core language
  - Boolean constraints & features
- **Today:** Advanced requirements

```
features
    Pizza
        mandatory
            Dough {Calories 10, Type 'Wheat'}
            Cheese {Calories 5}
            Sauce
                alternative
                    Tomato {Calories 2}
                    Pesto {Calories 4}
        optional
            Mushrooms {Calories 1}
            Ham {Calories 7}
            Pineapple {Calories 2}
            "Greetings on box"

constraints
    Pineapple => Ham
```

Complex Requirements

```
features
    Pizza
        mandatory
            Dough {Calories 10, Type 'Wheat'}
            Cheese {Calories 5}
            Sauce
                alternative
                    Tomato {Calories 2}
                    Pesto {Calories 4}
        optional
            Mushrooms {Calories 1}
            Ham {Calories 7}
            Pineapple {Calories 2}
            "Greetings on box"

constraints
    Pineapple => Ham
```

Language Levels Motivation



VS

+ Simple
+ Easy to understand
– Limited applicability

+ Covers more use cases
– Complex
– Harder to understand

Language Levels Boolean

- Boolean constraints & features
- Feature attributes for information

```
features
    Pizza
        mandatory
            Dough {Calories 10, Type 'Wheat'}
            Cheese {Calories 5}
            Sauce
                alternative
                    Tomato {Calories 2}
                    Pesto {Calories 4}
        optional
            Mushrooms {Calories 1}
            Ham {Calories 7}
            Pineapple {Calories 2}
            "Greetings on box"

constraints
    Pineapple => Ham
```

Language Levels Arithmetic

- Numeric constraints over feature attributes
- Expressions such as ==

```
include
    Arithmetic.*

features
    Pizza
        mandatory
            Dough {Calories 10, Type 'Wheat'}
            Cheese {Calories 5}
            Sauce
                alternative
                    Tomato {Calories 2}
                    Pesto {Calories 4}
        optional
            Mushrooms {Calories 1}
            Ham {Calories 7}
            Pineapple {Calories 2}
            "Greetings on box"

constraints
    Pineapple => Ham
    sum(Calories) < 28
```

Language Levels Type

- Feature types
  - Real
  - Integer
  - String
- Constraints over typed features

```
include
    Arithmetic.*
    Type.*

features
    Pizza
        mandatory
            Dough {Calories 10, Type 'Wheat'}
            Integer Cheese {Calories 5, Unit 'g'}
            Sauce
                alternative
                    Tomato {Calories 2}
                    Pesto {Calories 4}
        optional
            Mushrooms {Calories 1}
            Ham {Calories 7}
            Pineapple {Calories 2}
            String "Greetings on box"

constraints
    Pineapple => Ham
    sum(Calories) < 28
    Cheese < 300
    len("Greetings on box") < 100
```

Language Levels The Pain

```
include
    Arithmetic.*

features
    Pizza
        mandatory
            Dough {Calories 10, Type 'Wheat'}
            Cheese {Calories 5}
            Sauce
                alternative
                    Tomato {Calories 2}
                    Pesto {Calories 4}
        optional
            Mushrooms {Calories 1}
            Ham {Calories 7}
            Pineapple {Calories 2}
            "Greetings on box"

constraints
    Pineapple => Ham
    sum(Calories) < 28
```

```
features
    Pizza
        mandatory
            Dough {Calories 10, Type 'Wheat'}
            Cheese {Calories 5}
            Sauce
                alternative
                    Tomato {Calories 2}
                    Pesto {Calories 4}
        optional
            Mushrooms {Calories 1}
            Ham {Calories 7}
            Pineapple {Calories 2}
            "Greetings on box"

constraints
    Pineapple => Ham
```

```
include
    Arithmetic.*
    Type.*

features
    Pizza
        mandatory
            Dough {Calories 10, Type 'Wheat'}
            Integer Cheese {Calories 5, Unit 'g'}
            Sauce
                alternative
                    Tomato {Calories 2}
                    Pesto {Calories 4}
        optional
            Mushrooms {Calories 1}
            Ham {Calories 7}
            Pineapple {Calories 2}
            String "Greetings on box"

constraints
    Pineapple => Ham
    sum(Calories) < 28
    Cheese < 300
    len("Greetings on box") < 100
```

Language Levels The Pain

```
include
    Arithmetic.*

features
    Pizza
        mandatory
            Dough {Calories 10, Type 'Wheat'}
            Cheese {Calories 5}
            Sauce
                alternative
                    Tomato {Calories 2}
                    Pesto {Calories 4}
        optional
            Mushrooms {Calories 1}
            Ham {Calories 7}
            Pineapple {Calories 2}
            "Greetings on box"

constraints
    Pineapple => Ham
    sum(Calories) < 28
```

```
features
    Pizza
        mandatory
            Dough {Calories 10, Type 'Wheat'}
            Cheese {Calories 5}
            Sauce
                alternative
                    Tomato {Calories 2}
                    Pesto {Calories 4}
        optional
            Mushrooms {Calories 1}
            Ham {Calories 7}
            Pineapple {Calories 2}
            "Greetings on box"

constraints
    Pineapple => Ham
```

```
include
    Arithmetic.*
    Type.*

features
    Pizza
        mandatory
            Dough {Calories 10, Type 'Wheat'}
            Integer Cheese {Calories 5, Unit 'g'}
            Sauce
                alternative
                    Tomato {Calories 2}
                    Pesto {Calories 4}
        optional
            Mushrooms {Calories 1}
            Ham {Calories 7}
            Pineapple {Calories 2}
            String "Greetings on box"

constraints
    Pineapple => Ham
    sum(Calories) < 28
    Cheese < 300
    len("Greetings on box") < 100
```

Conversion Strategies

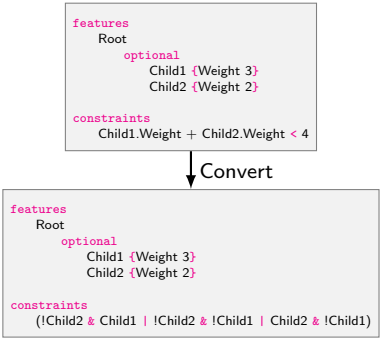
Language Levels

Boolean: `Pineapple => Ham`

Arithmetic: `sum(Calories) < 28`

Type: `Integer Cheese`

Conversion Strategies



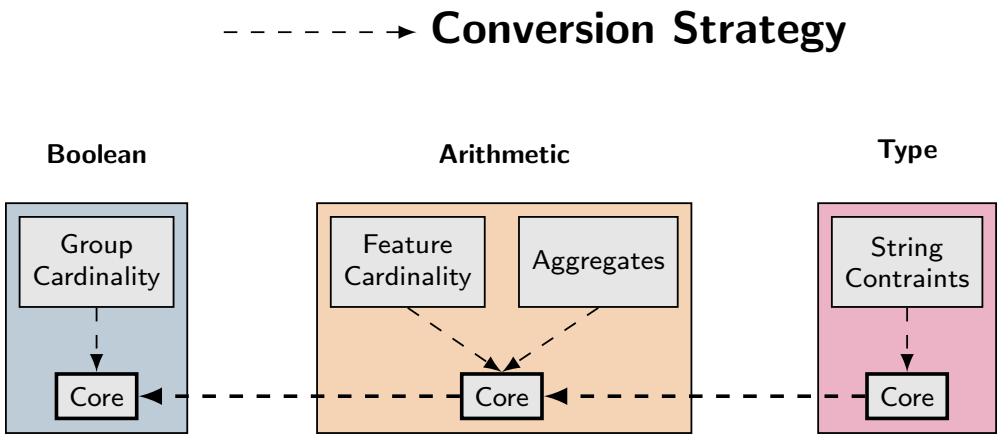


Figure: Language Levels in UVL

Conversion Strategies Realization

```
features
    Root
        optional
            Child1 {Weight 3}
            Child2 {Weight 2}

constraints
    Child1.Weight + Child2.Weight < 4
```

Invalid Input

Conversion Strategies Realization

```
features
    Root
        optional
            Child1 {Weight 3}
            Child2 {Weight 2}

constraints
    Child1.Weight + Child2.Weight < 4
```

Conversion
UVLParser

```
features
    Root
        optional
            Child1 {Weight 3}
            Child2 {Weight 2}

constraints
    (!Child2 & Child1 | !Child2 & !Child1 | Child2 & !Child1)
```

Conversion Strategies Realization

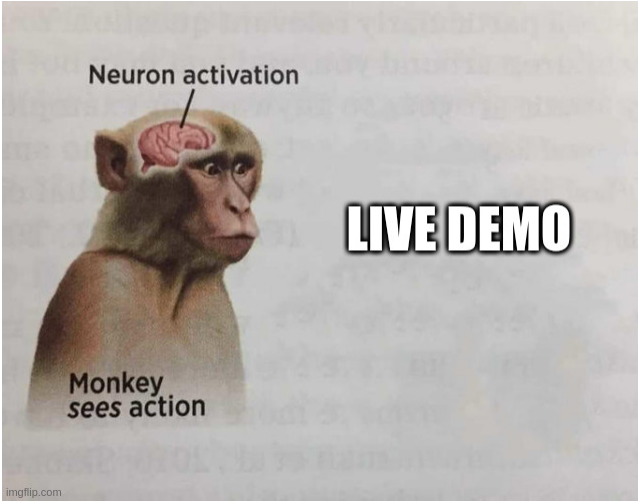
```
features
    Root
        optional
            Child1 {Weight 3}
            Child2 {Weight 2}

constraints
    Child1.Weight + Child2.Weight < 4
```

Conversion
UVLParser

```
features
    Root
        optional
            Child1 {Weight 3}
            Child2 {Weight 2}

constraints
    (!Child2 & Child1 | !Child2 & !Child1 | Child2 & !Child1)
```

Valid Input

Demo Time!


Neuron activation
LIVE DEMO
Monkey
sees action
imgflip.com

UVLParser: Extending UVL with Language Levels and Conversion Strategies

- Java-based
- UVL Parser
- Syntax checks
- Conversion of language levels

Figure: UVL Parser GitHub Repository

UVLParser: Extending UVL with Language Levels and Conversion Strategies

1. **Universal Variability Language**
2. **Language Levels**
3. **Conversion Strategies**