2P-Kt: logic programming with objects & functions in Kotlin

Giovanni Ciatto[*] Roberta Calegari[°] Enrico Siboni[†]
Enrico Denti[‡] Andrea Omicini[⋆]

[*‡⋆]Dipartimento di Informatica – Scienza e Ingegneria (DISI)
[°]Alma Mater Research Institute for Human-Centered Artificial Intelligence
Alma Mater Studiorum—Università di Bologna, Italy
{giovanni.ciatto, roberta.calegari, enrico.denti, andrea.omicini}@unibo.it

[†]University of Applied Sciences and Arts of Western Switzerland (HES-SO)
enrico.siboni@hevs.ch

21st Workshop "From Objects to Agents" (WOA)
Sept. 16, 2020, Bologna, Italy

Next in Line...

Context

AI side

- AI is shining, brighter than ever
  - mostly thanks to the advances in ML and sub-symbolic AI

$\Rightarrow$ symbolic AI is gaining momentum because of XAI

! hybrid solution mixing logic & data-driven AI are flourishing [2]

MAS side

The MAS community is eager for logic-based technologies [1]

- to support agents' knowledge representation, reasoning, or execution
- or to prove MAS properties

! despite few mature tech exist, and even fewer are actively maintained

Motivation

The problem with logic-based technologies

There is technological barrier slowing

- the adoption of logic programming (LP) as paradigm
- the exploitation of logic-based technologies

while programming *in the large*

- mainstream programming languages (*e.g. Scala, Kotlin, Python, C#*) are blending several paradigms
  - e.g. imperative, object-oriented (OOP), and functional programming (FP)
    - except LP!
- mainstream platforms (*e.g. JVM, .NET, JS, Python*) are poorly interoperable with logic-based tech.

Motivating example – SWI-Prolog's FLI for Java

- Many Prolog implementors rely on Foreign Language Interfaces (FLI)
  - (mostly targetting Java, or C)
- For instance, SWI-Prolog comes with a FLI for Java:

```
Query query = new Query("parent", new Term[] {
      new Atom("adam"),
      new Variable("X")
    }
  );  // ?- parent(adam, X).
Map<String,Term> solution = query.oneSolution();
System.out.println("The child of Adam is " + solution.get("X"));
```

→ No paradigm harmonization between Prolog and the hosting language

i.e. Java

Contribution of the paper

Explicitly state the contributions of the paper

- contribution 1
- contribution 2

Next in Line...

Theory / modelling / design

Provide 2-3 slides discussing the Theory / modelling / design

Next in Line...

Case study / Experiments / Results

Provide 2-3 slides discussing the Case study / Experiments / Results of the paper

Next in Line...

Conclusions & future works

Summing up

Summarise the most relevant contributions of this study:

- conclusion 1
- conclusion 2
- conclusion 3

Future works

Sketch some future research directions

- future work 1
- future work 2

(may be split into 2 slides)

2P-Kt: logic programming with objects & functions in Kotlin

Giovanni Ciatto[*] Roberta Calegari[°] Enrico Siboni[†]
Enrico Denti[‡] Andrea Omicini[⋆]

[*‡⋆]Dipartimento di Informatica – Scienza e Ingegneria (DISI)
[°]Alma Mater Research Institute for Human-Centered Artificial Intelligence
Alma Mater Studiorum—Università di Bologna, Italy
{giovanni.ciatto, roberta.calegari, enrico.denti, andrea.omicini}@unibo.it

[†]University of Applied Sciences and Arts of Western Switzerland (HES-SO)
enrico.siboni@hevs.ch

21st Workshop "From Objects to Agents" (WOA)
Sept. 16, 2020, Bologna, Italy

References I

[1] Roberta Calegari, Giovanni Ciatto, Viviana Mascardi, and Andrea Omicini.
Logic-based technologies for multi-agent systems: A systematic literature review.
Autonomous Agents and Multi-Agent Systems, In press.
Special Issue "Current Trends in Research on Software Agents and Agent-Based Software Development".

[2] Roberta Calegari, Giovanni Ciatto, and Andrea Omicini.
On the integration of symbolic and sub-symbolic techniques for XAI: A survey.
Intelligenza Artificiale, In press.