Policy-Making Environment

Mostafa Mohajeri
University of Amsterdam

DL4LD meeting
juni 16, 2022

My Role in DL4LD

- A computational environment for **policy-making**
  - **Design and test policies by model execution**
- Policies are always about a system
  - Computational model of Policies
  - Computational model of the System
- No assumption about policy model tool/language/framework
  - Focus on modelling the system subjected to policies

Modelling Systems

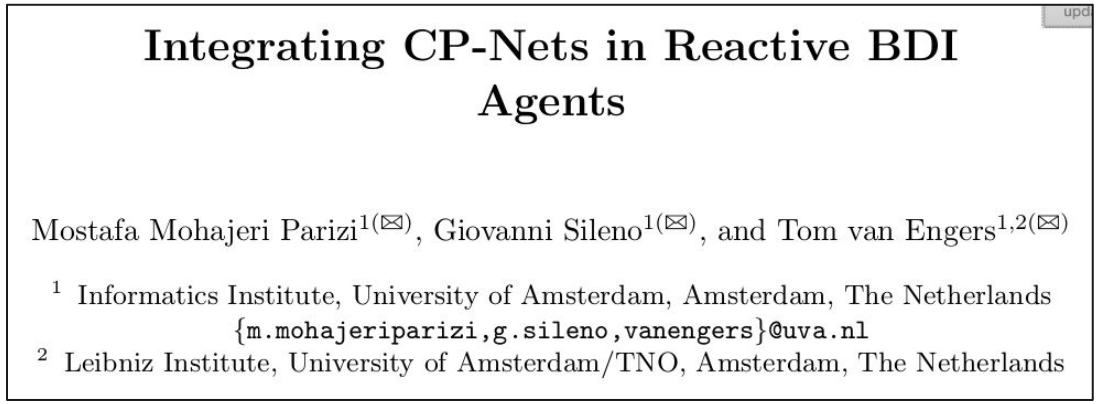
- Systems at hand are socio-technical
  - Both Software and Social actors
- Requirements in modelling social actors
  - Expressive and transparent decision-making
  - Reason with and about norms
- Requirements in modelling software actors
  - Interoperability
    - Often it is the real system, or, has the same interface
    - Technology matters! (Specially in data-sharing)

Agent-Oriented Programming to Model Actors

- AgentScript Cross-Compiler (ASC2)
  - A Multi-Agent System (MAS) framework
  - Social actors are modelled as **agents**
  - Software actors can be modelled as any executable entity

Expressive and Transparent Decision-Making

- ASC2 uses an Agent-Oriented Programming (AOP) language
  - Designed to be close to human mental attitudes
  - Formal and verifiable (under some constraints)
  - Focus on preference reasoning



Integrating CP-Nets in Reactive BDI Agents

Mostafa Mohajeri Parizi[1(✉)], Giovanni Sileno[1(✉)], and Tom van Engers[1,2(✉)]

[1] Informatics Institute, University of Amsterdam, Amsterdam, The Netherlands
{m.mohajeriparizi,g.sileno,vanengers}@uva.nl
[2] Leibniz Institute, University of Amsterdam/TNO, Amsterdam, The Netherlands

Declarative Preferences in Reactive BDI Agents

Mostafa Mohajeri Parizi[1], Giovanni Sileno[1], and Tom van Engers[1]

Complex Cyber Infrastructure, University of Amsterdam, the Netherlands
{m.mohajeriparizi,g.sileno,vanengers}@uva.nl

Preference-Based Goal Refinement in BDI Agents

Mostafa Mohajeriparizi
University of Amsterdam
Amsterdam, The Netherlands
m.mohajeriparizi@uva.nl

Giovanni Sileno
University of Amsterdam
Amsterdam, The Netherlands
g.sileno@uva.nl

Tom van Engers
University of Amsterdam
Amsterdam, The Netherlands
t.m.vanengers@uva.nl

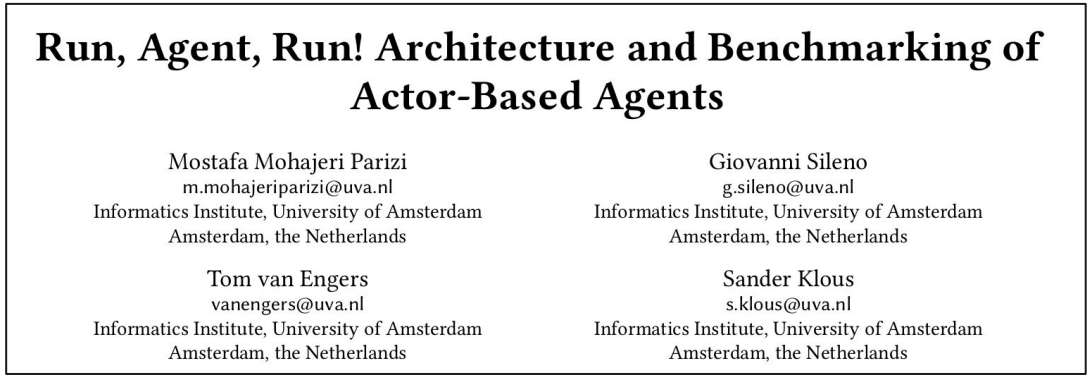
Interoperability

- ASC2 is a cross-compiler
  - Translates AOP language into Scala/Java
  - The same Interoperability as any JVM program + Integration of GraalVM
  - Directly compatible with development, DevOps or CI/CD tools

Seamless Integration and Testing for MAS Engineering

Mostafa Mohajeri Parizi[1], Giovanni Sileno[1], and Tom van Engers[1]

Informatics Institute, University of Amsterdam, Amsterdam, the Netherlands
{m.mohajeriparizi,g.sileno,t.m.vanengers}@uva.nl



Run, Agent, Run! Architecture and Benchmarking of Actor-Based Agents

Mostafa Mohajeri Parizi
m.mohajeriparizi@uva.nl
Informatics Institute, University of Amsterdam
Amsterdam, the Netherlands

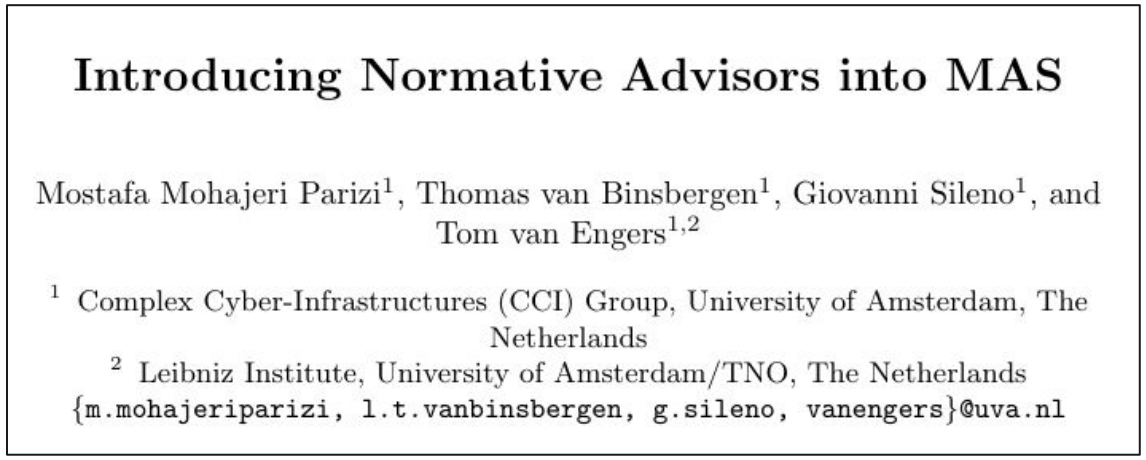
Giovanni Sileno
g.sileno@uva.nl
Informatics Institute, University of Amsterdam
Amsterdam, the Netherlands

Tom van Engers
vanengers@uva.nl
Informatics Institute, University of Amsterdam
Amsterdam, the Netherlands

Sander Klous
s.klous@uva.nl
Informatics Institute, University of Amsterdam
Amsterdam, the Netherlands

Reasoning with and about norms

- The framework allows for agents to
  - Adopt and drop norms
  - Follow or violate norms
  - Coordinate based on norms
- Completely independent of the used norms/policy framework



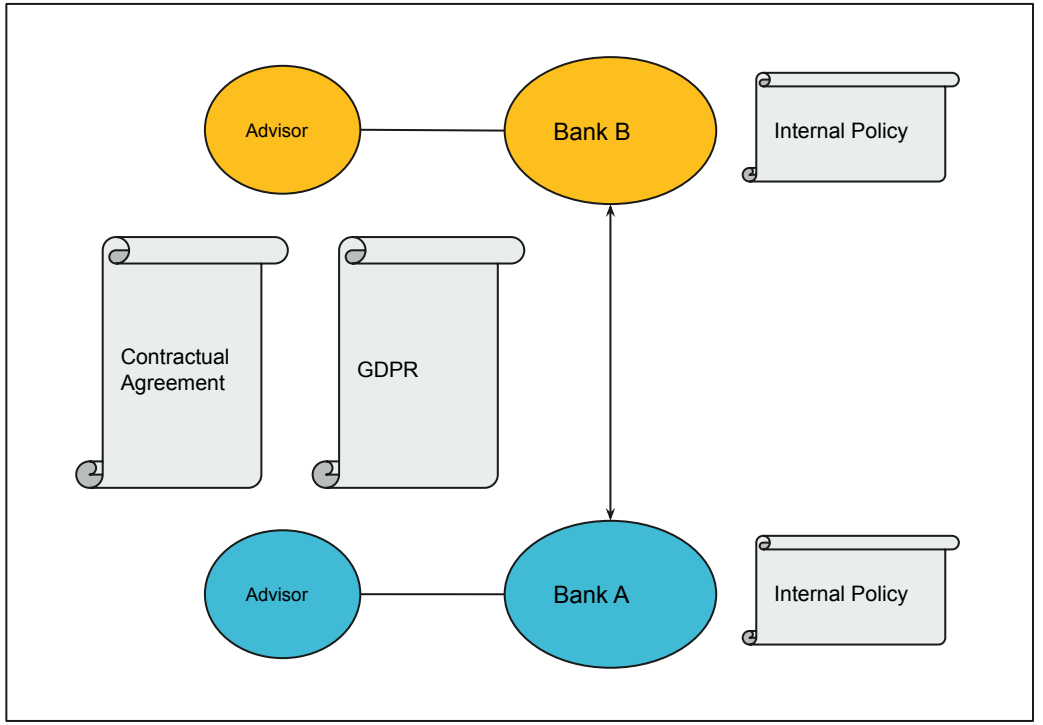
Introducing Normative Advisors into MAS

Mostafa Mohajeri Parizi[1], Thomas van Binsbergen[1], Giovanni Sileno[1], and Tom van Engers[1,2]

[1] Complex Cyber-Infrastructures (CCI) Group, University of Amsterdam, The Netherlands
[2] Leibniz Institute, University of Amsterdam/TNO, The Netherlands
{m.mohajeriparizi, l.t.vanbinsbergen, g.sileno, vanengers}@uva.nl

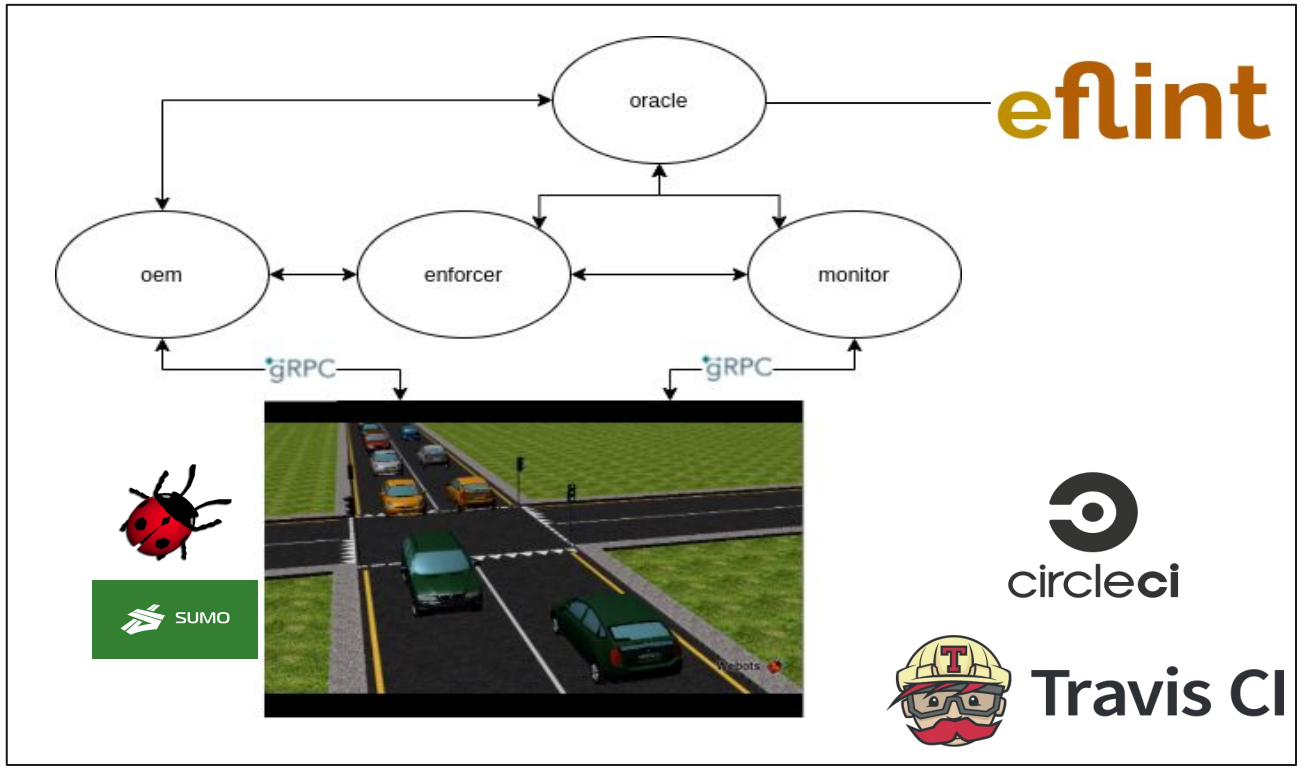
Case Study: KYC and GDPR in Bank Data-Sharing

- **K**now **Y**our **C**ustomer (KYC) and GDPR in Banking Systems' Data-Sharing



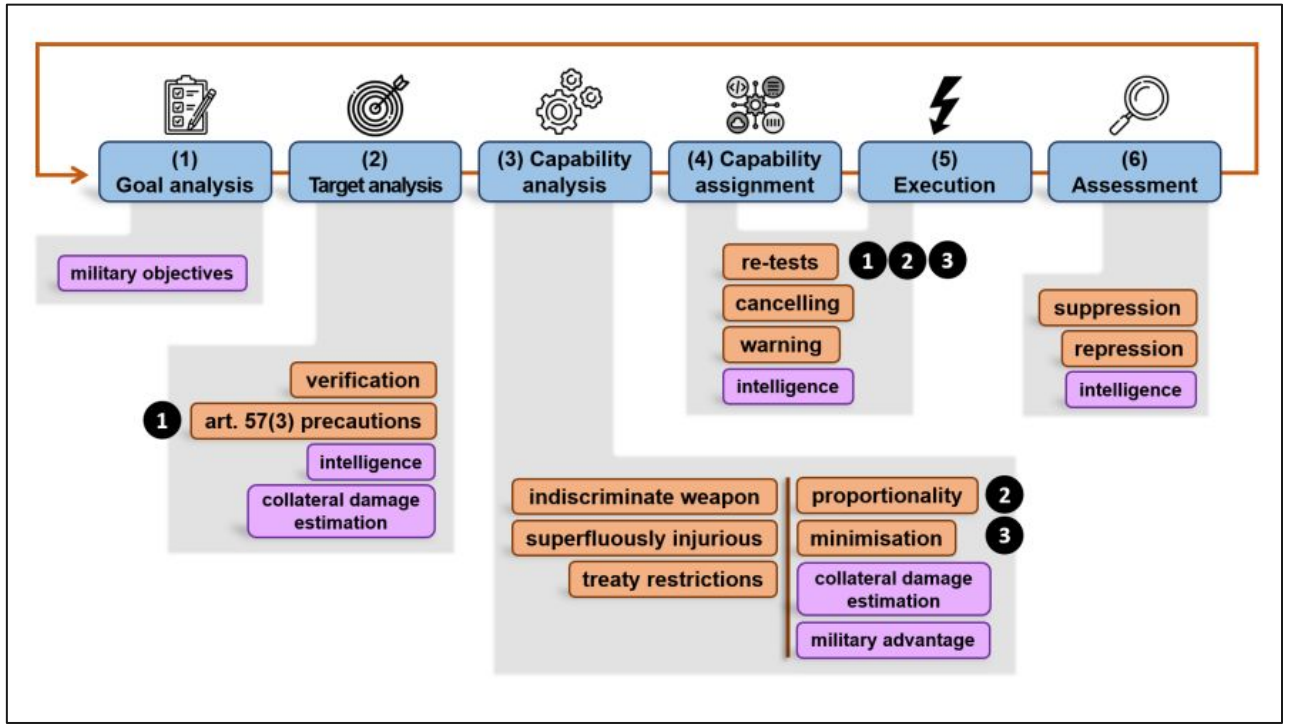
Case Study: LICCAM

- **LICCAM: L**egal **I**nterventions for **C**onnected and **C**ooperative **A**utomated **M**obility
  - (TNO)



Case Study: IHL

- Integrating IHL Into Autonomous Devices
  - (Asser Institute)



Conclusions

- Policy design is a complex process
  - Many complications can only be found with model execution
- Policy design process is akin to software design process
  - They should happen together for the best outcome
  - **Without getting in each other's way**
- But … software development tools are much more advanced
  - Version control, DevOps, IDEs, test libs, etc.
- Policy and Software tools should be seamlessly integratable
  - Need to create better tools!

Policy-Making Environment

Mostafa Mohajeri
University of Amsterdam

DL4LD meeting
Jun 16, 2022