On the Formal Verification of Open Multi-agent Systems

F. Belardinelli[1]

[1]Laboratoire IBISC
Université d'Evry

joint work with D. Grossi and A. Lomuscio

LAMAS SING – 23 September 2015

Overview

1. Background:
   - plenty of work on model checking Multi-agent Systems [LQR09, GvdM04, KNN$^+$08]:
     1. MAS are composed of a **finite number** of agents given at design time ...
     2. and they are described at **propositional level** (CTL, LTL, ATL, + epistemics, etc.)
2. Main task: **formal** verification of **open** MAS
   - given a model $\mathcal{M}_S$ of system S and a formula ϕ_P for property P, does $\mathcal{M}_S \models \phi_P$?
   - **open:** agents can enter and leave the MAS at run-time [JMS13]
     - model checking is appropriate for control-intensive applications...
     - ...but less suited for data-intensive applications (data typically range over infinite domains) [BK08]
3. Motivation:
   - auctions, markets, etc.
   - (non-probabilistic) diffusion phenomena (how information, ideas, behaviors spread in networks of agents similarly to epidemics)
     - SIR model for epidemics
   - Social Network Analysis (SNA) [Jac08, EK10]
4. Key contribution:
   - verification of **open** MAS is decidable ...
   - ... whenever the system is **bounded**
   - application to the case study – SIR model for epidemics

The **SIR** Model

- Influential network diffusion model [EK10, Jac08]
- Individuals are liable to go through three different stages during an epidemic:
  - first, each agent is **susceptible** to be infected
  - she may actually get **infected** at a certain point
  - finally she will eventually **recover**
- Verifiable behaviours:
  1. every agent either remains susceptible or will eventually become infected if she is continuously in contact with someone infected
  2. if an agent **knows** that she is connected to **some infected agent**, then she will part at some point in the future
  3. if an agent gets infected, then **all agents** that are connected to her will eventually **know** this fact.
- Results:
  - (non-stochastic) SIR model can be captured within open MAS
  - specifications such as (1)-(3) above can be (expressed and) model-checked

Challenges & Research Questions

Challenges:

- Multi-agent System, but ...
- ...the number of agents is **potentially infinite**
- the system is **open**: agents can join in or leave at run-time
- states have a **relational structure**
- the state space is **infinite** in general!

  $\Rightarrow$ the model checking problem cannot be tackled by standard techniques.

Research questions:

1. is the verification of open MAS decidable?
2. if not, can we identify **relevant** fragments that are reasonably well-behaved?

Open Multi-agent Systems

Technical Results

1. **Open Multi-agent Systems** (OMAS) as a flexible and rich framework for SNA.

   Intuition: encoding an agent's information structure as a database.

2. FO-CTLK$_x$ as a specification language:

$$\forall x, y(K_x(Inf(y) \wedge N(x,y)) \rightarrow AF\neg N(x,y))$$

   if an agent knows that she is connected to some infected agent, then she will part at some point in the future

   - epistemic operators indexed to terms in the language
   - quantification on those indexes

3. We leverage on recent results on data-aware systems to tackle model checking [BPL14, HCG$^+$13, MCD14].

   Main result: abstraction techniques to reduce the MC problem to the finite case.

4. Case study: modelling and verification of the SIR model.

Data-aware Systems

Preliminaries on databases

- Recent paradigm in Service-Oriented Computing [CH09].

- Motto: let's give **data** and **processes** the same relevance!
  - the data content shapes the actions of processes

- Agents' local states are represented as databases.
  - a database schema is a **finite** set $\mathcal{D} = \{P_1/q_1, \ldots, P_n/q_n\}$ of relation symbols P_i with arity $q_i \in \mathbb{N}$
  - a (database) instance on a domain U is a mapping D associating each symbol P_i with a **finite** q_i-ary relation on U
  - the active domain $adom(D)$ is the set of all elements $u \in U$ appearing in some $D(P_i)$
  - the disjoint union $D \oplus D'$ of $\mathcal{D}$-instances D and D' is the $(\mathcal{D} \cup \mathcal{D}')$-instance s.t.
    - ⋆ $D \oplus D'(P) = D(P)$
    - ⋆ $D \oplus D'(P') = D'(P)$
  - $\mathcal{D}(U)$ is the set of all $\mathcal{D}$-instances on U

- Intuition: networks (graphs on agents) are represented as first-order structures

Open Multi-agent Systems
Agents

Hereafter we assume

- a finite number of **agent types** $T_0, \ldots, T_k$
  - as well as a **possibly infinite** set Ag_T of agent names for each type T
  - the interpretation domain U includes $Ag = \bigcup_{type\ T} Ag_T$

Definition (Agent)

An agent $a_T = \langle \mathcal{D}_T, Act_T, Pr_T \rangle$ of type T

- records information according to the local database schema $\mathcal{D}_T$
  - including a dedicated unary predicate N to represent the network structure
- and performs the actions $\alpha(\vec{x})$ in Act_T ...
- ...according to the local protocol function $Pr_T : \mathcal{D}_T(U) \mapsto 2^{Act_T(U)}$

- the number of agent types is **finite**:
  ⇒ typically it is possible to specify the relevant agent types at design time.
- the number of agents is **infinite**:
  - it is much more difficult to know how many agents of each type will appear during the system's execution.
- the setting is reminiscent of the **interpreted system semantics** for MAS [FHMV95], ...
  ...but here the local state of each agent is relational.

Example: the SIR Model I

In the basic setting we have a unique type of agent.

- the interpretation domain is $U = Ag$.
- an agent a includes
  - a local db schema
    $$\mathcal{D}_a = \{Sus/1, Inf/1, Rec/1, N/1\}$$
  - a set of actions
    $$Act_a = \{con(ag), disc(ag), \text{skip}\}$$
  - the protocol Pr_a is such that
    - $disc(b) \in Pr_a(l_a)$ whenever $b \in l_a(N)$
    - $\{\text{skip}, con(b)\} \subseteq Pr_a(l_a)$ for all $l_a \in \mathcal{D}_a(U)$

We might want to assess the impact of health workers on epidemics.

- we consider a new type T_H and set Ag_H of agent names
- a health worker h has database $\mathcal{D}_h$ and actions Act_h defined as for standard agents.
  - while the protocol Pr_h is such that
    - $disc(b) \in Pr_h(l_h)$ only if $b \in l_h(N)$ and $Inf(h) \in l_h$

The framework is rich enough to accommodate several versions of the SIR model.

Open Multi-agent Systems

OMAS

Agents interact, thus generating OMAS.

Definition (Global State)

Given a **finite** subset $A \subseteq Ag$ of agents $a_i = \langle \mathcal{D}_i, Act_i, Pr_i \rangle$, for $i \leq n$, a global state is a tuple $s = \langle l_0, \ldots, l_n \rangle$ of instances $l_i \in \mathcal{D}_i(U)$.

- at every state only **finitely many** agents are active
  - if $s = \langle l_{a_0}, \ldots, l_{a_n} \rangle$ then $ag(s) = \{a_0, \ldots, a_n\}$ is the set of agents active in s
- key difference w.r.t. interpreted (parametric) systems: global states may be tuples of different lengths

Definition (OMAS)

An OMAS $\mathcal{P} = \langle Ag, U, I, \rightarrow \rangle$ describes

- the evolution of a **possibly infinite** group Ag of agents ...
- from an initial global state $s_0 \in I$ ...
- according to the transition relation $s \xrightarrow{\alpha(\vec{u})} s'$
  - where $\alpha(\vec{u})$ contains an action for each agent active in s

OMAS are infinite-state systems in general

Example: the SIR Model II

The **SIR OMAS** $\mathcal{P} = \langle Ag \cup Ag_H, I, \tau \rangle$ **with health workers** is defined as

- I is the set of states where at least one agent is infected (this rules out trivial models).
- $\rightarrow$ is the transition relation s.t. $s \xrightarrow{\alpha(\vec{u})} s'$ whenever
  - a susceptible agent a might get infected if she is in contact with an infected agent: if $Sus(a) \in l_a$ and for some $b \in l_a(N)$, $Inf(b) \in l_b$, then either $Sus(a) \in l'_a$ or $Inf(a) \in l'_a$
  - an infected agent a non-deterministically recovers: if $Inf(a) \in l_a$, then either $Inf(a) \in l'_a$ or $Rec(a) \in l'_a$
  - a recovered agent a does not fall ill again: if $Rec(a) \in l_a$ then $Rec(a) \in l'_a$
  - the consistency of the agents' information is assumed to be preserved.
  - ...

The Specification Language: FO-CTLK$_x$

- First-order version of CTL + knowledge:

$$\varphi \quad ::= \quad R(t_1, \dots, t_c) \mid t = t' \mid \neg\varphi \mid \varphi \to \varphi \mid \forall x \varphi \mid AX\varphi \mid A\varphi U \varphi \mid E\varphi U \varphi \mid K_a \varphi \mid K_x \varphi$$

  Epistemic operators indexed to terms in the language.

- OMAS $\mathcal{P}$ **satisfies** formula φ in state s for assignment σ, iff

$$\begin{array}{lll}
(\mathcal{P}, s, \sigma) \models R(\vec{t}) & \text{iff} & \langle \sigma(t_1), \dots, \sigma(t_c) \rangle \in D_s(R) \\
(\mathcal{P}, s, \sigma) \models t = t' & \text{iff} & \sigma(t) = \sigma(t') \\
(\mathcal{P}, s, \sigma) \models \forall x \varphi & \text{iff} & \text{for all } u \in adom(s),\ (\mathcal{P}, s, \sigma^x_u) \models \varphi \\
(\mathcal{P}, s, \sigma) \models AX\varphi & \text{iff} & \text{for all runs } r,\ r(0) = s \text{ implies } (\mathcal{P}, r(1), \sigma) \models \varphi \\
(\mathcal{P}, s, \sigma) \models A\varphi U \varphi' & \text{iff} & \text{for all runs } r,\ r(0) = s \text{ implies } (\mathcal{P}, r(k), \sigma) \models \varphi' \text{ for some } k \geq 0, \\
 & & \text{and } (\mathcal{P}, r(k'), \sigma) \models \varphi \text{ for all } 0 \leq k' < k \\
(\mathcal{P}, s, \sigma) \models E\varphi U \varphi' & \text{iff} & \text{there exists } r \text{ s.t. } r(0) = s,\ (\mathcal{P}, r(k), \sigma) \models \varphi' \text{ for some } k \geq 0, \\
 & & \text{and } (\mathcal{P}, r(k'), \sigma) \models \varphi \text{ for all } 0 \leq k' < k \\
(\mathcal{P}, s, \sigma) \models K_a \varphi & \text{iff} & \text{for all states } s',\ s \sim_a s' \text{ implies } (\mathcal{P}, s', \sigma) \models \varphi \\
(\mathcal{P}, s, \sigma) \models K_x \varphi & \text{iff} & \text{for all states } s',\ s \sim_{\sigma(x)} s' \text{ implies } (\mathcal{P}, s', \sigma) \models \varphi
\end{array}$$

  where $s \sim_a s'$ iff $a \in ag(s)$, $a \in ag(s')$, and $s_a = s'_a$.

- Active-domain semantics, but...
  - ...we can refer to individuals that no longer exist
  - the number of states is infinite in general

The Specification Language: FO-CTLK$_x$

1. each agent goes through the susceptible-infected-recovered cycle

$$\forall x A(Sus(x) U A(Inf(x) U Rec(x)))$$

2. if an agent knows that she is connected to some infected agent, then she will part at some point in the future

$$\forall \mathbf{x}, y(K_{\mathbf{x}}(Inf(y) \wedge N(\mathbf{x}, y)) \rightarrow AF \neg N(\mathbf{x}, y))$$

3. if an agent gets infected, then all agents that are connected to her will eventually know this fact.

$$\forall y(Inf(y) \rightarrow (AF \forall \mathbf{x}(N(\mathbf{x}, y) \rightarrow K_{\mathbf{x}} Inf(y))))$$

- $\forall x K_x \phi$ expresses dynamically the joint knowledge of ϕ for all active agents in a given state, i.e., the standard, static epistemic formula $E\phi = \bigwedge_{a \in Ag} K_a \phi$.
- epistemic formulas are vacuously true for agents not in the active domain of the state considered:
  - $a \notin ag(s)$ implies $(P, s, \sigma) \models K_a \phi$ for all formulas ϕ

Verification of AC-MAS

- Model-checking problem: given
  - an OMAS $\mathcal{P}_S$ (for a system S)
  - an FO-CTLK$_x$ sentence ϕ_P (representing property P)

  we check that

$$\mathcal{P}_S \models \phi_P$$

- Problem: the infinite domain U may generate infinitely many states!

- Investigated solution: can we **simulate** the concrete values in U with a finite set of **abstract** symbols?

Abstraction: Isomorphism and Bisimulation

- two states s, s' are isomorphic, or $s \simeq s'$, if **they share the same relational structure**

	$D(R)$	
A_1	a	b
A_2	b	c
A_3	d	e

$\simeq$

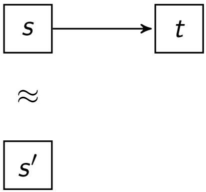
	$D'(R)$	
A_1	1	2
A_2	2	3
A_3	4	5

- i.e., there is a bijection $\iota : adom(s) \cup ag(s) \mapsto adom(s') \cup ag(s')$ such that
  - ι preserves the type of agents
  - for every tuple $\vec{u}$ and agent $a_i \in ag(s)$,

$$\vec{u} \in D_i(P) \Leftrightarrow \iota(\vec{u}) \in D'_{\iota(i)}(P)$$

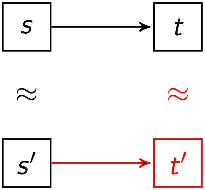
Abstraction: Isomorphism and Bisimulation

- two states s, s' are bisimilar, or $s \approx s'$, if
  1. $s \simeq s'$
  2. the simulation and transition relations commute



Abstraction: Isomorphism and Bisimulation

- two states s, s' are bisimilar, or $s \approx s'$, if
  1. $s \simeq s'$
  2. the simulation and transition relations commute



  - if $s \to t$ then there is t' s.t. $s' \to t'$, $s \oplus t \simeq s' \oplus t'$, and $t \approx t'$
  - the other direction holds as well
  - similar conditions hold for the epistemic relation $\sim_a$

Uniformity

- the behaviour of OMAS is **independent** from data not explicitly named in the system description.

s	
a	b
b	c
d	e

$\longrightarrow$

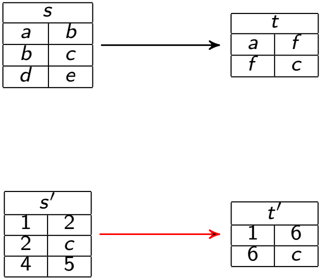
t	
a	f
f	c

s'	
1	2
2	c
4	5

t'	
1	6
6	c

Uniformity

- the behaviour of OMAS is **independent** from data not explicitly named in the system description.



- OMAS are **uniform**:
  - for $s, t, s' \in \mathcal{S}$ and $t' \in \mathcal{D}(U)$, $s \to t$ and $s \oplus t \simeq s' \oplus t'$ imply $s' \to t'$
- Uniformity holds in many cases of interest [CH09, BPL14, HCG+13, MCD14].

Bisimulation and Equivalence w.r.t. FO-CTLK$_x$

Bisimilar OMAS satisfy the same FO-CTLK$_x$ formulas (provided some assumption on the cardinalities of the domains)

Theorem

Consider

- **bisimilar** *OMAS* $\mathcal{P}$ *and* $\mathcal{P}'$
- *an FO-CTLK$_x$ formula* φ

If

1. $|U'| \geq 2 \cdot \sup_{s \in \mathcal{P}}\{|adom(s) \cup ag(s)|\} + |vars(\varphi)|$
2. *for every type* T, $|Ag'_T| \geq 2\sup_{s \in \mathcal{P}}\{|ag_T(s)|\} + |vars(\varphi)|$
3. $|U| \geq 2 \cdot \sup_{s' \in \mathcal{P}'}\{|adom(s') \cup ag(s)|\} + |vars(\varphi)|$
4. *for every type* T, $|Ag_T| \geq 2\sup_{s' \in \mathcal{P}'}\{|ag_T(s')|\} + |vars(\varphi)|$

then

$$\mathcal{P} \models \varphi \quad \textit{iff} \quad \mathcal{P}' \models \varphi$$

Can we apply this result to obtain finite abstraction?

Bounded Models and Finite Abstractions

- an OMAS $\mathcal{P}$ is b-bounded iff for all $s \in \mathcal{P}$, $|adom(s) \cup ag(s)| \leq b$.
- bounded systems can still be infinite!

Theorem

Consider

- *a b-**bounded** OMAS $\mathcal{P}$ on an infinite domain U*
- *an FO-CTLK$_x$ formula φ*

*Given a **finite** domain U' s.t.*

1. $|U'| \geq 2b + \max\{|vars(\varphi)|, b \cdot N_{Ag}\}$
2. *for every type T,* $|Ag'_T| \geq 2b + \max\{|vars(\varphi)|, b \cdot N_{Ag}\}$

*there exists a **finite abstraction** $\mathcal{P}'$ of $\mathcal{P}$ s.t. $\mathcal{P}'$ is bisimilar to $\mathcal{P}$.*
In particular,

$$\mathcal{P} \models \varphi \quad \textit{iff} \quad \mathcal{P}' \models \varphi$$

⇒ Under specific circumstances (namely boundedness), we can model check an infinite-state OMAS by verifying its finite abstraction.

- Boundedness is a natural assumption on the SIR model.
  - For a sufficiently large b, we can simulate a b-bounded SIR model with a domain U' s.t. $|U'| = 3b$.

Conclusions

- Results:
  - bisimulation and finite abstraction for open Multi-agent Systems
  - we are able to model check OMAS w.r.t. FO-CTLK$_x$ ...
  - ... however, our results hold only for **bounded** systems
  - this class covers many interesting systems (AS programs, [CH09, HCG$^+$11, BPL14])
  - including the SIR model

- Future Work:
  - constructive techniques for finite abstraction
  - model checking techniques for finite-state systems are effective on OMAS?
  - how to perfom the boundedness check?

Questions?

References

Christel Baier and Joost-Pieter Katoen.
Principles of Model Checking (Representation and Mind Series).
The MIT Press, 2008.

Francesco Belardinelli, Fabio Patrizi, and Alessio Lomuscio.
Verification of agent-based artifact systems.
Journal of Artificial Intelligence Research, 51:333–77, 2014.

D. Cohn and R. Hull.
Business Artifacts: A Data-Centric Approach to Modeling Business Operations and Processes.
IEEE Data Eng. Bull., 32(3):3–9, 2009.

D. Easley and J. Kleinberg.
Networks, Crowds, and Markets.
Cambridge University Press, 2010.

R. Fagin, J.Y. Halpern, Y. Moses, and M.Y. Vardi.
Reasoning About Knowledge.
The MIT Press, 1995.

P. Gammie and R. van der Meyden.
MCK: Model checking the logic of knowledge.
In *Proceedings of 16th International Conference on Computer Aided Verification (CAV'04)*, volume 3114 of *LNCS*, pages 479–483. Springer-Verlag, 2004.

B. Bagheri Hariri, D. Calvanese, G. De Giacomo, R. De Masellis, and P. Felli.
Foundations of Relational Artifacts Verification.
In *Proc. of BPM*, 2011.

B. Bagheri Hariri, D. Calvanese, G. De Giacomo, A. Deutsch, and M. Montali.
Verification of relational data-centric dynamic systems with external services.
In R. Hull and W. Fan, editors, *PODS*, pages 163–174. ACM, 2013.

M. O. Jackson.
Social and Economic Networks.