Normative rational agents – a BDI approach

Mihnea Tufiş
Jean-Gabriel Ganascia

Université Pierre et Marie Curie Paris 6
Laboratoire d'Informatique de Paris 6

Outline

1. About norms and normative MAS
2. Testing scenario – a SF novel
3. State of the Art
4. Our Approach – normative BDI agents
5. Implementing the normative BDI agent
6. Future Work
7. Conclusions
8. Q&A

Norms
General

The Merriam-Webster dictionary:

- an authoritative standard
- a principle of right action binding upon the members of a group and serving to guide, control and regulate proper and acceptable behavior

- a pattern or trait taken to be typical in the behavior of a social group
- a widespread or usual practice, procedure, or custom

Norms
More technically

- Regulation or pattern of behavior meant to prevent an excess in the autonomy of an agent

- Examples:
  - One should wait for others to get off the bus, before getting on
  - Household robots should not care for babies, except in emergencies *[McCarthy, 2001]*

Normative multi-agent systems

- Normchange definition: MAS + set of norms
  - agents: decide to follow explicitly represented norms
  - normative set: how can an agent modify the norms

  [Boella et al., 2006]

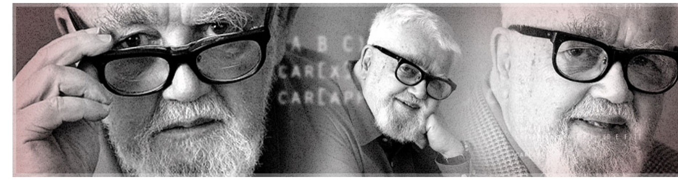
- Mechanism change definition: MAS organized by means of mechanisms to:
  - represent, communicate, distribute, detect, create, modify, enforce norms
  - detect norm violations and norm fulfillment

  [Boella et al., 2007]

Research Questions

- How do we formally represent a norm?
- When does a norm become active? What happens when a norm contradicts other norms or the rational states of an agent? How do we solve such conflicts?
- How does an active norm become part of the agent's mental model?

Source: innovation.it.uts.edu.au/projectjmc/articles/robotandbaby.html



Testing scenario

The Robot and the Baby (2001), by Prof. John McCarty

Source: http://www.scenicreflections.com

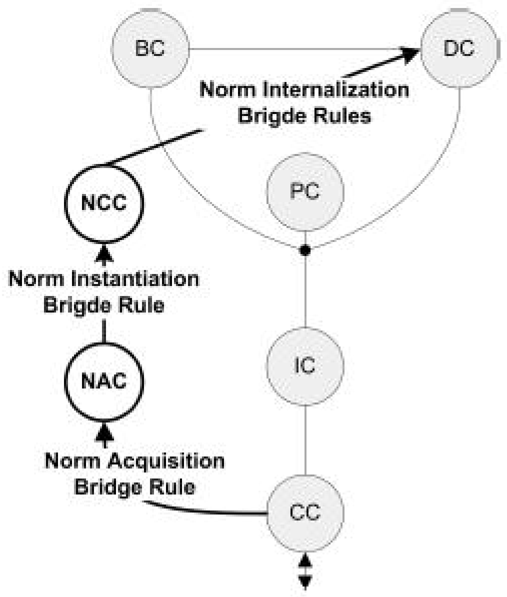
State of the Art
NoA

- Why useful?
  - Relevant research questions: norm adoption, norm consistency
  - Consistency check
- Limits:
  - Considers only a *reactive* agent architecture
  - No consistency check against mental states (doesn't really have any!)

[Kollingbaum et al., 2007]

State of the Art
A BDI architecture for norm compliance

- Why useful?
  - Context-based architecture
  - Norm formalization



- Limits:
  - No support for consistency check
  - No details about the impact on the BDI execution loop

[Criado et al., 2010]

Our Approach
Outline

- Representing norms
- The “classical“ BDI agent
- The normative BDI agent
  - Norm acceptance
  - Norm instantiation
  - Conflict detection and conflict resolution
  - Norm internalization

Representing norms
Abstract norm

- ***Abstract* norm**: n_a = <M, A, E, C, R, S>
  - M = F / P / O : prohibition / permission / obligation
  - A, E : activation / expiration conditions
  - C : activity regulated by the norm
  - R, S : reward / sanction

  [Criado et al., 2010]

- Examples:

  (F, love(R781,Travis), none, none, x, y)

  (O, feed(R781,Travis), health(Travis)<0.2, health(Travis)>0.5, x, y)

Representing norms
Norm instance

- **Norm *instance***: n_i = <M, C'>
  - Given belief theory Γ_{BC} and n_a:
    - Γ_{BC} |- σ(A)
    - C' = σ(C), where σ / A s.t. σ(A), σ(E), σ(S), σ(R) grounded

  [Criado et al., 2010]

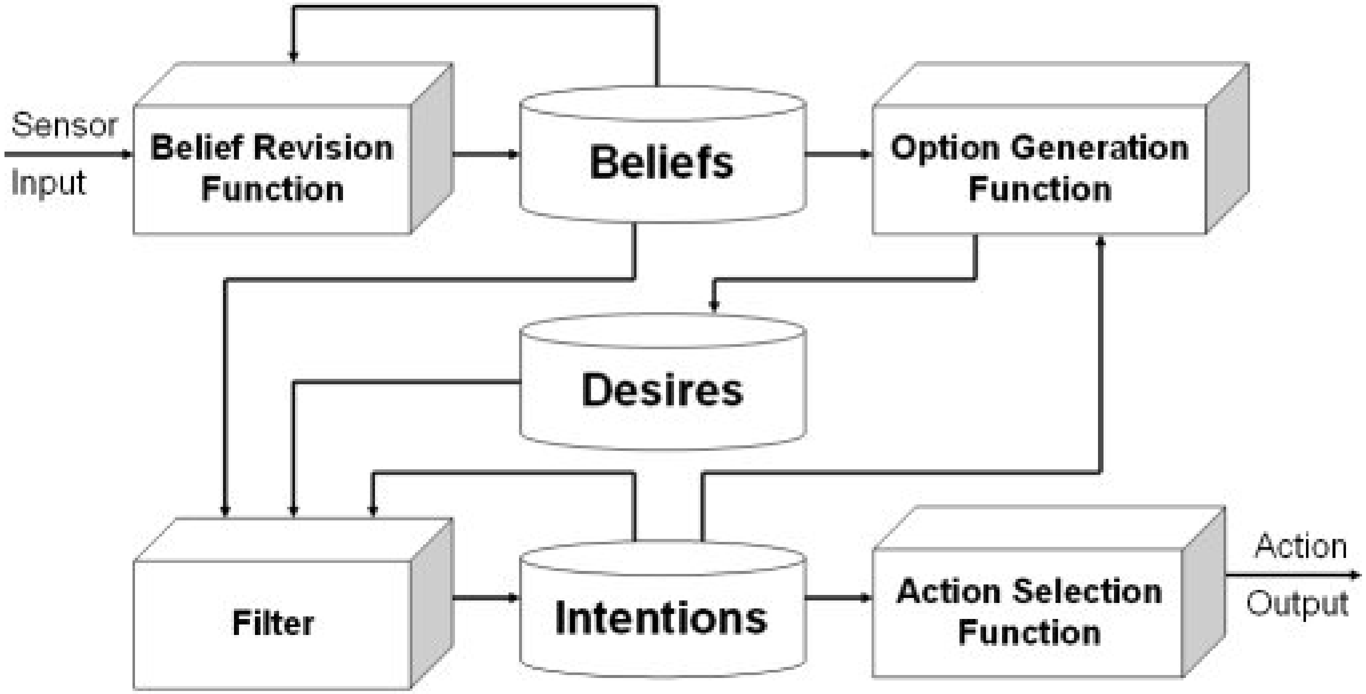
- Example:

  Γ_{BC} = {…, health(Travis) = 0.1, …}

  n_a= (O, feed(R781,Travis), health(Travis)<0.2, health(Travis)>0.5, x, y)

  n_i= (O, feed(R781,Travis))

BDI Agent Architecture
Recall



[Wooldridge, 2009]

The normative BDI agent
Architecture

- Mental context
  - belief-set, desire-set, intention-set
- Normative context
  - storing abstract norms
  - storing norm instances
- Bridge rules
  - norm instantiation bridge rule
  - norm internalization bridge rule
- Consistency module
  - consistency check
  - solving conflicts

Norm instantiation
Accepting a norm

- Abstract Norm Base (ANB)
  - stores in-force norms (not yet accepted by an agent!)
- Norm Instance Base (NIB)
  - stores active norms (accepted by an agent)
  - acceptance is done only after consistency is checked
- Norm instantiation bridge rule

  ANB: <M, A, E, C, R, S>
  Bset: B(A), B(¬E)
  \-
  NIB: <M, C’>

Testing Scenario
Formalization

ANB: -

NIB: <F, love(R781,Travis)>

Bset: <B, ¬healthy(Travis)>
<B, hungry(Travis)>

<B, csq(¬love(R781,x)) $>_c$ csq(heal(R781, x))>

Dset: <D, ¬love(R781, Travis)>
<D, healthy(Travis)>

Iset: -

```
PLAN heal(x,y)
{
    pre:  ¬healthy(y)
    post: healthy(y)
    Ac:   feed(x,y)
}
```

```
PLAN feed(x,y)
{
    pre:   ∃x.love(x,y) & hungry(x)
    post: ¬hungry(x)
}
```

Norm instantiation
Example

- New abstract norm:

  <O, love(R781,Travis), none, none, x, y>

- Norm instance:

  <O, love(R781,Travis)>

Consistency check

New obligation vs. Existing norms

$$consistent(p, NIB) \iff (effects(n_i^F) \setminus effects(n_i^P)) \cap effects(p) = \emptyset \\ \wedge \\ effects(n_i^O) \cap neg\_effects(p) = \emptyset$$

$$strong\_inconsistency(o, NIB) \iff \forall p \in options(o).(\exists \langle F, p \rangle \in NIB \wedge \nexists \langle P, p \rangle \in NIB) \\ \vee \\ \neg consistent(p, NIB)$$

$$strong\_consistency(o, NIB) \iff \forall p \in options(o).\neg(\exists \langle F, p \rangle \in NIB \wedge \nexists \langle P, p \rangle \in NIB) \\ \wedge \\ consistent(p, NIB)$$

$$weak\_consistency(o, NIB) \iff \exists p \in options(o).(\exists \langle F, p \rangle \in NIB \wedge \nexists \langle P, p \rangle \in NIB) \\ \wedge \\ consistent(p, NIB)$$

Consistency check
New obligation vs. Mental attitudes

$$consistent(p, I) \iff \forall i \in I.(effects(\pi_i) \cap effects(p)) = \emptyset$$

$$strong\_inconsistency(o, I) \iff \forall p \in options(o).\neg consistent(p, I)$$

$$strong\_consistency(o, I) \iff \forall p \in options(o).consistent(p, I)$$

$$weak\_consistency(o, I) \iff \exists p \in options(o).consistent(p, I)$$

Conflict resolution

- Possible actions set: *P*
- Conflict set: Π(B, D) subset of *P*
- Maximal non-conflicting subset: φ
  - φ subset of Π, w/o conflicts
  - for all other φ' subset of Π, for which φ is a subset of φ', φ' has conflicts
- More than one maximal non-conflicting subsets?
  - select the actions which have the **least worse consequences**

[Ganascia, 2012]

Conflict resolution
Example

- Conflict set:
  - {love(R781, Travis), feed(R781, Travis), heal(R781, Travis), ¬love(R781, Travis)}
- Maximal non-conflicting subsets:
  - {love(R781, Travis), feed(R781, Travis), heal(R781, Travis)}
  - {¬love(R781, Travis)}
- Consequential value:
  - $csq(\neg love(x, y)) >_c csq(heal(x, y))$
- Resulting actions:
  - {love(R781, Travis), feed(R781, Travis), heal(R781, Travis)}

Norm internalization

- Newly acquired norms which are consistent become part of the agent's mental attitudes
- Ongoing debate about which attitudes should be updated, considering a new active norm

- Norm internalization bridge rules:

NIB: <O, C1>	NIB: <F, C2>
--------------------	---------------------
Dset: <D, C1>	Dset: <D, ¬C2>

Norm internalization
Example

- NIB:

  <O, love(R781, Travis)>

- Dset:

  <D, love(Travis)>

Implementation
Outline

- Jadex
  - agent development platform based on: agent theory, object-oriented programming, XML standard
  - BDI kernel

- System architecture
  - agent specification: ADF
  - norms specification: XML
  - plans specification: Java

Source: http://jadex-agents.informatik.uni-hamburg.de

Future work

- Norm acquisition
  - norm imitation
  - machine learning techniques
- Coherency check of normative and mental contexts
  - Thagard's coherence theory
  - coherence graphs
- Testing real life scenarios (Carte Vitale)
- Adapting the agent implementation using ASP (answer set programming)

Conclusions

- Investigated previous approaches on normative agents (reactive and rational)
- Adopted a formalization for defining norms
- Drawn from the nBDI architecture in order to adapt norms to a BDI agent
- Formalized consistency check (vs. norms and vs. mental attitudes)
- Provided with a conflict solving technique based on maximal non-conflicting sets and a consequentialist approach
- Jadex implementation of the normative BDI agent
- A challenging testing scenario, based on a SF novel

Thank you!

Jean-Gabriel.Ganascia@lip6.fr
tufism@poleia.lip6.fr

Questions…

Source: http://www.clipartof.com

References

1. G. Boella, L. van der Torre, H. Verhaegen, ‘Introduction to normative multiagent systems’, Computation and Mathematical Organizational Theory, Special issue on Normative Multiagent Systems, 12(2-3), 71–79, (2006).
2. Guido Boella, Gabriella Pigozzi, and Leendert van der Torre, ‘Normative systems in computer science - ten guidelines for normative multiagent systems’, in Normative Multi-Agent Systems, eds., Guido Boella, Pablo Noriega, Gabriella Pigozzi, and Harko Verhagen, number 09121 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, (2009). Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany.
3. Guido Boella, Leendert van der Torre, and Harko Verhagen, ‘Introduction to normative multiagent systems’, in Normative Multi-agent Systems, eds., Guido Boella, Leon van der Torre, and Harko Verhagen, number 07122 in Dagstuhl Seminar Proceedings, (2007).
4. Natalia Criado, Estefania Argente, Pablo Noriega, and Vicente J. Botti, ‘Towards a normative bdi architecture for norm compliance.’, in MALLOW, eds., Olivier Boissier, Amal El Fallah-Seghrouchni, Salima Hassas, and Nicolas Maudet, volume 627 of CEUR Workshop Proceedings. CEUR-WS.org, (2010).
5. Jean-Gabriel Ganascia, ‘An agent-based formalization for resolving ethical conflicts’, Belief change, Non-monotonic reasoning and Conflict resolution Workshop - ECAI, Montpellier, France, (August 2012).
6. Martin J. Kollingbaum and Timothy J. Norman, ‘Norm adoption and consistency in the noa agent architecture.’, in PROMAS, eds., Mehdi Dastani, Jrgen Dix, and Amal El Fallah-Seghrouchni, volume 3067 of Lecture Notes in Computer Science, pp. 169–186. Springer, (2003).
7. John McCarthy, ‘The robot and the baby’, (2001).
8. Anand S. Rao and Michael P. Georgeff, ‘Bdi agents: From theory to practice’, in In Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95, pp. 312–319, (1995).
9. Michael Wooldridge, An Introduction to MultiAgent Systems, Wiley Publishing, 2nd edn., 2009.