Norms and Electronic Institutions for Behaviour Regulation in Distributed Systems.

Applications to eContracting Environments

Javier Vázquez-Salceda

April 1, 2009

Contents

- Introduction
- A Language for Norms
- Normative Agents
- Norms and Agent Platforms: Electronic Institutions
- Contract-based Institutions
- Conclusions and Challenges
Introduction

Nowadays, computing trends move toward distributed solutions. Computer systems are networked into large distributed systems; processing power can be introduced in almost any place and device → processing becomes ubiquitous.

The agent paradigm is one way to conceptualize and implement distributed (intelligent) systems.

- Agents are human-oriented abstractions.
- Each agent can specialize in some (sub)problems and take decisions locally.
- Solutions to coordinate the agent society can be borrowed from human organizations and human societies.
Introduction (II)

- “An Intelligent Agent is a computer system that is capable of flexible, autonomous action on behalf of its user or owner”
- “By flexible we mean reactive, pro-active and social” [M. Wooldridge]

- Other desired properties: rationality, learning/adaptation.
  - Agents should be able to adapt their behavior to new, unexpected situations
- A Multiagent System (MAS) consists of a number of agents, interacting with one-another
  - It is desirable that agents in a MAS coordinate their behaviour and collectively adapt to unforeseen events
- Problem: how can we meet all these expectatives?

Introduction (III)

- Autonomy is one of the most desired properties of agents. We want agents to be autonomous in order to be able to (proactively) take their own decisions and to adapt to new, unexpected situations.
- We want agents to behave as expected, in order to achieve one or several goals. Therefore some control should be applied to the agents' behaviour.

- Agent Autonomy VS Control: problem:
  - How to ensure (control) an efficient and acceptable behaviour of a Multiagent System without diminishing the agents' autonomy?
Introduction (IV)

- **Norms** are a flexible way to specify the boundaries of acceptable (legal) behaviour
  - They specify WHAT is acceptable and WHAT is not, but not HOW
  - Agents have autonomy to reach their goals as far as they "move" within the acceptable boundaries.

- Norms ease agent interaction:
  - reduce uncertainty of other agents' behaviour
  - reduce misunderstanding in interaction
  - allows agents to foresee the outcome of an interaction
  - simplify the decision-making (reduce the possible actions)

- To ensure acceptable behaviour, a safe environment is needed: **Electronic Institutions**
  - Safe agent interaction environments
  - They include definition of norms and enforcement mechanisms

---

**Normative MAS: state of the Art (I)**
Normative MAS: state of the Art (II)

- **Normative Level**
  - Concrete
  - Abstract
  - Rules
  - [O, P, F]
  - [E, G, H]
  - Deliberative Normative Agents

- **Operational Level**
  - Procedural
  - Concrete
  - Abstract
  - 1 Ag. - 2 Ag.
  - EIDE, JADEX, JACK
  - TROPOS, EIDE/AMELI

Gap between Normative and Operational

- **Normative Level**
  - Laws, regulations

- **Operational Level**
  - EIDE
  - dialogical perspective
  - Dialogical Framework
  - Agent roles
  - Performative structure
  - Scenes
  - Conversational graphs
Example: Organ and Tissue Distribution

Abstraction problem

- Problems:
  - Norms are more abstract than the procedures (in purpose)
  - Norms do not have operational semantics

Example:

Regulation: "It is forbidden to discriminate potential recipients of an organ based on their age (race, religion, ...)"

Formal norm: \( F(\text{discriminate}(x,y,\text{age})) \)

Procedure: does not contain action "discriminate"
Filling the gap

too abstract and vague

more concrete

Laws, regulations

Language for norms (Formal & Computational)

Electronic Institutions

Norm enforcement mechanisms

Normative Agents

Norms in deliberation cycle

Normative Description (Deontic, Formal)

Operational Description (Operational, Computational)

Design guidance, Traceability

Maintenance

Operational Description

Normative Agents

Norms in deliberation cycle

Electronic Institutions

Norm enforcement mechanisms
A Language for Norms

Representing Norms (I)

- Formal representation of norms needed
- Which logic?
  - Norms permit, oblige or prohibit
  - Norms may be conditional
  - Norms may have temporal aspects
  - Norms are relativized to roles
- The representation should be easily parseable and usable by agents

variant of Deontic Logic
Representing Norms (II)

- Type 1: Unconditional norms about predicates
  - the norms on the value of $P$ are active at all times:
    
    $\text{OBLIGED}(a, P) \quad \text{PERMITTED}(a, P) \quad \text{FORBIDDEN}(a, P)$

  - an example:
    
    $\text{FORBIDDEN}(\text{recipient, (in\_waiting\_list(hospital))} \land$
    
    $\text{in\_waiting\_list(hospital2)} \land (\text{hospital1} \neq \text{hospital2}))$

- Type 2: Unconditional norms about actions
  - the norms on the execution of $A$ are active at all times:
    
    $\text{PERMITTED}(a \text{ DO } A) \quad \text{FORBIDDEN}(a \text{ DO } A)$

  - an example:
    
    $\text{FORBIDDEN}(\text{person DO sell(organ)})$

Representing Norms (III)

- Type 3: Conditional norms
  - the activation of the norms is conditional under $C$
  - $C$ may be a predicate about the system or the state of an action:
    
    $\text{OBLIGED}(a, P \text{ IF } C) \quad \text{OBLIGED}(a \text{ DO } A \text{ IF } C)$
    
    $\text{PERMITTED}(a, P \text{ IF } C) \quad \text{PERMITTED}(a \text{ DO } A \text{ IF } C)$
    
    $\text{FORBIDDEN}(a, P \text{ IF } C) \quad \text{FORBIDDEN}(a \text{ DO } A \text{ IF } C)$

  - an example:
    
    $\text{FORBIDDEN}(\text{allocator DO assign(organ, recipient)})$
    
    $\text{IF NOT(hospital DONE ensure\_quality(organ))}$
Representing Norms (IV)

- **Type 4: Conditional norms with Deadlines**
  - the activation of norms is defined by a deadline
    
    OBLIGED([a, P] BEFORE D)
    PERMITTED([a DO A] AFTER D)
    FORBIDDEN([a, P] BEFORE D)
  
    - absolute and relative deadlines:
      
      23:59:00 09/05/2004  \( \text{time(done(assign(organ, recipient)))} + 5\text{min} \)
  
    - an example:
      
      OBLIGED([allocator DO assign(heart, recipient)])
      BEFORE (time(done(extraction(heart, donor))) + 6hours))

Representing Norms (V)

- **Type 5: Obligations of enforcement of norms**
  - norms concerning agent b generate obligations on agent a:
    
    OBLIGED([a ENFORCE(OBLIGED(b...))])
    OBLIGED([a ENFORCE(PERMITTED(b...))])
    OBLIGED([a ENFORCE(FORBIDDEN(b...))])
  
    - an example:
      
      OBLIGED([ONT ENFORCE(FORBIDDEN(person DO sell(organ)))]))
Norms and Agents

Some domains (e.g. Medicine) are very sensible. We must ensure proper behaviour of agents. Agents should keep a certain autonomy.

We can express agents’ acceptable behaviour with norms.

Normative Agents (I)
Ensuring proper agent behaviour with norms:

- Some domains (e.g. Medicine) are very sensible
  - We must ensure proper behaviour of agents
  - Agents should keep a certain autonomy

- We can express agents’ acceptable behaviour with norms
Normative Agents (II)

- **Problem 1**: Which is the relation between the norms and the agents beliefs, desires and intentions?
- **Problem 2**: How exactly can norms define acceptable behaviour?
- **Idea**: We should first analyse the impact of norms on cognitive agents from a theoretical perspective.

Normative Agents (III)

- Our norms are expressed in deontic logic with proper Kripke semantics
  - Kripke model of the impact of norms
  - Possible worlds
- Our model is composed by 2 dimensions
  - *Epistemic dimension* (states and behaviours as Possible Worlds)
  - *Normative dimension* (norms applying to the agent)
Normative Agents (IV)

Safety and Soundness
- The concept of legally accessible worlds allows to describe:
  - wanted (legal) and unwanted (illegal) behaviour
  - acceptable (safe) and unacceptable (unsafe) states
- Violations when agents breaks one or more norms, entering in an illegal (unsafe) state.
- Sanctions are actions to make agents become legal (safe) again.
- Sanctions include the actions to recover the system from a violation.
Normative Agents (VI)

Context
- In real domains norms are not universally valid but bounded to a given context.
  - HC norms bounded to trans-national, national and regional contexts
- A Context is a set of worlds with a shared vocabulary and a normative framework
  - $e$-inst$_X$ is a context defining an ontology and a normative specification
- Usually nested contexts
  - There are super-contexts that have an influence in $e$-inst$_X$ ontology and norms
- Special impact on the Ontologies
  - Proposal: not to force a single representation for all contexts, but interconnected ontologies (multi-contextual ontologies).

Normative Agents (VII)
Implementing Normative Agents (I)

- **Problem:** HOW to introduce norms in the existing agent implementations?
- There are already implementations based in the BDI agent framework
  - E.g., 2APL agents, JACK agents, JADEx agents.
- **Idea:** Extend the BDI interpreter to include norms.

Plan selection

Plan execution

Agent Control Loop Version 7

1. $B := B_0$;
2. $I := I_0$;
3. while true do
4.   get next percept $\rho$;
5.   $B := br(h_0, \rho)$;
6.   $D := \text{options}(B, I)$;
7.   $I := \text{filter}(B, I, D)$;
8.   $\pi := \text{plan}(B, I)$;
9.   while not (completed or succeeded($I$) or impossible($I$)) do
10.     $\alpha := \text{kd}(\pi)$;
11.     execute($\alpha$);
12.     $\pi := \text{bd}(\pi)$;
13.     get next percept $\rho$;
14.     $B := br(h_0, \rho)$;
15.     if reconsider($I$, $B$) then
16.       $D := \text{options}(B, I)$;
17.       $I := \text{filter}(B, I, D)$;
18.     end-if
19.     if not sound($\pi, I, B$) then
20.       $\pi := \text{plan}(B, I)$;
21.     end-if
22. end-while
23. end-while

Norms and Agents (IX)

**Norm obligations**
- add actions to the set of options and may define some priorities or precedence

**Norm prohibitions**
- delete actions from the set of options

Problem: is not straight-forward in practice!
Norms in Agent Platforms:

Electronic Institutions

Electronic Institutions (I)

- Need of a safe environment where proper behaviour is enforced.
- Institutions are a kind of social structure where a corpora of constraints (the institution) shape the behaviour of the members of a group (the organization).
- An e-Institution is the computational model of an institution through the specification of its norms in (some) suitable formalism(s).

Hypothesis: Agent behaviour guided by Norms
Electronic Institutions (II)

- **Problem**: no connection between theoretical work on eInstitutions and practical implementations on eInstitutions
- First proposal: the HARMONIA framework
- Second proposal: the OMNI framework
- Third proposal (ongoing): the ALIVE framework
Electronic Institutions (III)
The OMNI framework

Abstract Level
- Statutes (values, objectives, context)
- Model Ontology

Concrete Level
- Norm level
  - Rule level
- Organizational Model
- Concreate Domain Ontology
- Generic Comm. Acts

Implementation Level
- Normative Implementation
- Social Model
  - Interaction Model
  - Agents
- Procedural Domain Ontology
  - Specific Comm. Acts

Electronic Institutions (III)
The OMNI framework

Normative Concrete Level
Organizational Model
- Role
  - Architectural Templates
  - Scene Norms
  - Transition Norms
  - Role Rules
  - Scene Rules
- Social structure
- Interaction structure

Ontological Concrete Level
- Ontologies
- Communication languages

Roles and institutions for Behaviour Regulation
- Statutes (values, objectives, context)
- Model Ontology
- Norm level
  - Rule level
- Organizational Model
- Concreate Domain Ontology
- Generic Comm. Acts
- Normative Implementation
- Social Model
  - Interaction Model
  - Agents
- Procedural Domain Ontology
  - Specific Comm. Acts
- Normative Dimension
- Organizational Dimension
- Ontological Dimension
Example:
From abstract to concrete in OMNI

LAWs

ABSTRACT LEVEL

$O_{ONT}(\text{appropriate(distribution)})$

CONCRETE LEVEL

$O_{ONT}(\text{ensure_appropriateness}($organ, recipient$) \prec \text{do}($assign($organ, recipient$)))$

$O_{CARRE} (\text{ensure_appropriateness}(\text{organ}, \text{recipient}) \prec \text{do}(\text{assign}(\text{organ}, \text{recipient})))$

$\text{assign}(\text{organ}, \text{recipient}) \text{ done}(\text{ensure_appropriateness}(\text{organ}, \text{recipient}))$

PROCEDURE LEVEL

Example:
Context as source of interpretation in OMNI

LAWs

ABSTRACT LEVEL

$O_{ONT}(\text{appropriate(distribution)})$

CONCRETE LEVEL

$O_{ONT}(\text{ensure_appropriateness}(\text{organ}, \text{recipient}) \prec \text{do}($assign($organ, recipient$)))$

$O_{CARRE} (\text{ensure_quality}(\text{organ}) \prec \text{do}(\text{assign}(\text{organ}, \text{recipient})))$

$O_{CARRE} (\text{ensure_compatibility}(\text{organ}, \text{recipient}) \prec \text{do}(\text{assign}(\text{organ}, \text{recipient})))$

$\text{assign}(\text{organ}, \text{recipient}) \text{ done}(\text{ensure_quality}($organ$))$

$\text{assign}(\text{organ}, \text{recipient}) \text{ done}(\text{ensure_compatibility}($organ, recipient$))$

PROCEDURE LEVEL
Adapting OMNI to AMELI

**LAW**

Concrete Level:

\[ OBLIGED(ONT, \text{appropriate}(|\text{distribution}|)) \]

Spanish National Health System

Abstract Level:

\[ OBLIGED(ONT, \text{ensure}_\text{appropriateness}(|\text{organ}, \text{recipient}|) \prec \text{do}(\text{assign}(|\text{organ}, \text{recipient}|))) \]

\[ OBLIGED(ONT, \text{ensure}_\text{quality}(|\text{organ}|) \text{ BEFORE } \text{do}(\text{assign}(|\text{organ}, \text{recipient}|))) \]

Procedure Level:

\[ OBLIGED(\text{utter}(|S7, W3, \text{quality}_\text{ensured}(|\text{organ}|)) \text{ IF } (\text{uttered}(|S7, W3, \text{assign}(|\text{organ}, \text{recipient}|)))) \]

\[ \text{uttered}(|S7, W3, \text{assign}(|\text{organ}, \text{recipient}|))^\lor \text{not } \text{uttered}(|S7, W3, \text{quality}_\text{ensured}(|\text{organ}|)) \]

AMELI Implementation

---

Implementing Norms in eInstitutions (I)

- Implementation of norms from institutional perspective
  - Implementing a theorem prover to check protocol compliance

- Implementation of a safe environment
  - (**norm enforcement**)

- 2 options depending on control over agents
  - Defining constraints on unwanted behaviour
  - Defining violations and reacting to these violations

**our assumptions:**

- Norms can be sometimes violated by agents
- The internal state of agents is neither observable nor controllable
  - actions cannot be imposed on an agent’s intentions
  - agents as black boxes
  - only their observable behaviour and actions
Implementing Norms in eInstitutions (II)

- **Norms** describe which states/actions within the e-organization should *ideally* take place
- **Norms** are too abstract, no operational
  - A norm implementation is composed by:

```
Norm FORBIDDEN(assign(organ, recipient))
condition IF NOT(assign(organ, recipient))

Violation NOT(done(ensure.quality(organ))) AND
condition done(assign(organ, recipient))

Detection detect_alarm(assign, starting);
institution check(done(ensure.quality(organ)));

Sanction inform(board, "NOT(done(ensure.quality(organ)))
AND done(assign(organ, recipient))");

Repair {stop assign(organ);
    record("NOT(done(ensure.quality(organ))) AND
    done(assign(organ, recipient))", incident.log);
    detect_alarm(ensure.quality, done);
    check(done(ensure.quality(organ)));
    resume assign(organ);}
```

---

**CONTRACT**

**SOA Governance**

**as**

**Contract-based Institutions**
Target: Service Oriented Architectures

- Current trend in distributed computation: **Webservices**, **GRID computing**
- Service Oriented Architectures framework
  - Broad definition of *service* as component that takes some inputs and produces some outputs.
  - Services are brought together to solve a given problem typically via a *workflow* definition that specifies their composition.
- Every application is made up of *actors*
- Every change that happens is an action by an actor
- Actors communicate by sending *messages*
- Every action is triggered by a message
- The outputs of (messages sent by) an actor are *caused* by the inputs to (messages received by) the actor

Direct mapping to multiagent systems

How can norm compliance be introduced in SOA?

- **SOA governance**
  - refers to policies and software tools that aim to manage service-oriented architecture
  - involves both design-time and run-time aspects
    - Design-time: enterprise architects create a set of rules that define
      - how services should be constructed
      - how services may be deployed (including access rights)
    - Run-time: Governance software
      - helps put the SOA guidelines into action
      - monitors the performance of services
- **SOA provenance**
  - Refers to desired process definition (workflows) and software tools to trace process execution
  - Includes tools to register meaningful events and interactions and to re-create
SOA and the ‘Future Internet’

- Visions of Service Oriented Business Environments are well established

- Huge challenges remain, in particular:
  - Greater scale and openness conflict with standard assumptions about the behaviour of actors in the world
  - Increased Autonomy / Flexibility conflict with our ability to ensure predictable execution
  - Dynamic discovery / late binding conflict with the need for Sound Legal Guarantees

- The gap between human perceptions of business interactions – and their low level implementation remains very large

Contract-based SOA Governance

- Contract based approaches promise two clear med/long term benefits in Service Oriented Business environments:
  - Closer linkage between technical implementation and responsibilities / obligations
  - Abstraction away from internal execution details in order to support formal verification of distributed enterprise systems

- Project Meme:
  - Contracts are a proxy / specification for action by business software components, they can provide the basis for sound specification of distributed business systems.
Where are the Contracts?

- **Contracts:**
  - Make explicit the obligations of each of the parties in the transactions
  - Make explicit what each system can expect from another

- **Bind together:**
  - The electronic interaction (web services) with
  - The business obligation with
  - Prediction as to whether the system will function to get the job done

- A contract instantiation creates a contracting environment
  - Monitors contractual clauses (deontic statements \(\rightarrow\) norms!)
  - The environment is, in fact, an electronic institution!

Contracting language overview (I)

**Contract expressions**

```xml
<ISTContract
  ContractName="AftercareContract"
  StartingDate="2007-01-01T00:00:00+01:00"
  EndingDate="2008-01-01T00:00:00+01:00"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://www.ist-contract.org/schemas/ISTContract.xsd">
  ...
  </Contextualization>
  ...
  </Definitions>
  ...
  </Clauses>
  ...
</ISTContract>
```
Contracting language overview (I)
Contract expressions

```xml
<ISTContract
   ContractName="AftercareContract"
   StartingDate="2007-01-01T00:00:00+01:00"
   EndingDate="2008-01-01T00:00:00+01:00"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="http://www.ist-contract.org/schemas/ISTContract.xsd">
   <Contextualization>
   ...
   </Contextualization>
   <Definitions>
   ...
   </Definitions>
   <Clauses>
   ...
   </Clauses>
</ISTContract>
```

```xml
<ISTContract
   ContractName="AftercareContract"
   StartingDate="2007-01-01T00:00:00+01:00"
   EndingDate="2008-01-01T00:00:00+01:00"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="http://www.ist-contract.org/schemas/ISTContract.xsd">
   <Contextualization>
   ...
   </Contextualization>
   <Definitions>
   ...
   </Definitions>
   <Clauses>
   ...
   </Clauses>
</ISTContract>
```
Contracting language overview (I)
Contract expressions

```xml
<ISTContract
  Contra
  Starting
  Ending
  xmlns:
  xsi:no?
  <Contextualization
    ...
  </Contextualization
  <Definitions
    ...
  </Definitions
  <Clauses
    ...
  </Clauses
</ISTContract>

OBLIGED (Operator
  DO PayForEngine(amount, engine, Operator, EngineManufacturer)
  BEFORE (2008-07-1T15:30:30+01:00)

  <booleanExpression>
    Before(2008-07-1T15:30:30+01:00)
  </booleanExpression>
  </ExplorationCondition>
  <DeonticStatement>
    <Modality>
      OBLIGATION
    </Modality>
    <Who>
      Operator
    </Who>
    <What>
      PayForEngine(amount, engine, Operator, EngineManufacturer)
    </What>
  </DeonticStatement>
</Clause>
```

Contracting language overview (II)
Relations between language components
Contracting language overview (III)
Communication Model

Interaction context:
Protocol handling:
Message envelope + intentionality: from service S1 to service S2 ...
Request[cancel(contract C1)]

Statements / actions related to contracts:
cancel(contract C1)

A contract:
"the workshop is obliged to repair the car in 2 days"

Domain terms: car, workshop, repair

Filling the gap

too abstract and vague
more concrete

Laws, regulations, Business rules
Electronic Contracts
Action Descriptions, Workflows
Contract-Aware Agents
(Clause) Norms in deliberation cycle
Contractual Institutions
(Clause) Norm enforcement mechanisms

Design guidance, Maintenance
Traceability
Conclusions

- New systems interconnected in distributed scenarios
  - E.g. Health Care services
- Need to explicitly handle the problem of
  - variety of regulations
  - trust, coordination and communication between agents of different systems
- Proposal of a language for norms
- Concept of Normative Agents.
  - Norms to define acceptable behaviour
  - Impact on the agent implementation
- Concept of Electronic Institutions
  - Norms to build a safe environment
  - Implementation of enforcement mechanisms
- Contracts as one way to bring institutions into SOA
  - Clauses are agreed norms between contractual parties
  - A contract instantiation creates an institution on-demand
Ongoing work (ALIVE): using landmarks for formal connection

- **Landmarks** as meaningful (i.e. important) states in the system
- **Landmark patterns**: partial accessibility relations from landmark to landmark
- Idea 1: do not try to map ALL states, only the landmarks
- Regulations usually define those important states, and what should/should never happen among them
  - We can define landmarks in the normative level in terms of acceptable/unacceptable states of affairs
  - We can define landmarks in the operational level as states in the state machine

- Hypothesis: an execution is norm-compliant if the landmark patterns hold.

---

**From Norms to Landmark Patterns**

\[
O(\rho \leq \delta) \\
F\psi
\]
From Landmark Patterns to Protocols/Workflows

\[ \psi \]

\[ \text{uttered}(S,W,F) \text{ IF } C \]

http://www.lsi.upc.es/~jvazquez