From Human Regulations to Regulated Software Agents’ Behaviour.

(elInstitutions: the KEMLG@UPC and IS@Utrecht view)

Javier Vázquez-Salceda

May 20, 2005

MOTIVATION
**The Role of Norms and Electronic Institutions**

**Elnstitutions and Norms**

- **Norms** describe which states/actions within the e-organization should ideally take place.
- **Norms** are too abstract to be directly translated into procedures (plans/protocols) in a single step.
- Most of the approaches talk about **norms**, but a close-up look shows that they are working at completely different **levels of abstraction**.
- **Idea**: there are **several levels of abstraction** involved in a normative system.
- Organizations hardly work in isolation.
- **Idea**: to identify how the organization’s surrounding **context** influences the different levels.

---

**State of the Art (I)**

[Diagram showing the interface between normative and descriptive levels, with theoretical and practical approaches, and different types of interactions.]
The role of Norms and Electronic Institutions...

Normative MAS: state of the Art (II)

Gap between Normative and Descriptive
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”
The role of Norms and Electronic Institutions... 

Filling the gap

Laws, regulations

Language for norms (Formal & Computational)

Electronic Institutions
Norm enforcement mechanisms

Normative Agents
Norms in deliberation cycle

more concrete

too abstract and vague

Operational Description (Operational, Computational)

Design guidance, Maintenance, Traceability

Normative Description (Deontic, Formal)
From ABSTRACT to CONCRETE

- **STATUTES**
  - Values
  - Objectives
  - Context

- **Operational Description**
  - Norms

- **Procedural Implementation**

---

**ONTOLOGIES**

- Generic Human Law Terms
- Context-specific Terms
- Context-specific Actions
- Communication ontologies
  - XML

---

**CONTEXT**

- Domain Regulations
- Domain Procedures
Problem 1: Abstraction in Norms

- Norms are **abstract** if they use concepts that are not fully described in the organization's ontology.
  - "It is forbidden to discriminate based on age"

- Norms can be **abstract** in the following ways:
  - They refer to an **abstract action**
  - They use **terms** that are **vague**
  - They abstract from **temporal aspects**
  - They abstract from **agents** and/or **roles**
  - They refer to actions or situations that are **not** (directly) **controllable** and/or **verifiable** by the organization

**Example 1: Abstract actions**

- "a living donor should consent to the donation of an organ"

\[
\text{sign(donor,contract)} \cup \text{carry(donor,will)} \cup \text{tell(donor,family)} \implies \text{ONT Consent(donor)}
\]

**Example 2: Vague terms**

- "the ONT is obliged to ensure that the distribution of organs and tissues is appropriate"

\[
\text{ONT(ensure\_quality(organ))} \land \\
\text{ONT(ensure\_compatibility(organ, recipient))} \implies \\
\text{ONT(appropriate(distribution))}
\]
From Normative to Operational

- Translation from Normative dimension to a Descriptive one
  - Idea: reduction from Deontic Logic to Dynamic Logic
    [J.-J. Meyer]

\[
\begin{align*}
O_{\text{hosp}}(\text{consent}(\text{donor}(p,x)) & < \text{do}(\text{transplant}(\text{hosp},x,p,q))) \\
\text{[transplant}(\text{hosp},x,p,q)) & \text{done}(\text{consent}(\text{donor}))
\end{align*}
\]

\[
\begin{align*}
O_{\text{buyer}}(\text{pay}(\text{goods},\text{seller},\text{price}) & < \text{do}(\text{exit}(\text{buyer}))) \\
\text{not}(\text{done}(\text{pay}(\text{goods},\text{seller},\text{price}))) & \rightarrow [\text{exit}(\text{buyer})] \lor (\text{fine}(\text{buyer}))
\end{align*}
\]

Example

- **ABSTRACT LEVEL**
  - LAWS
    - \(O_{\text{ONT}}(\text{appropriate}(\text{distribution}))\)

- **CONCRETE LEVEL**
  - \(O_{\text{ONT}}(\text{ensure}_{\text{appropriateness}}(\text{organ},\text{recipient}) < \text{do}(\text{assign}(\text{organ},\text{recipient})))\)
  - \(O_{\text{CARREL}}(\text{ensure}_{\text{appropriateness}}(\text{organ},\text{recipient}) < \text{do}(\text{assign}(\text{organ},\text{recipient})))\)

- **PROCEDURE LEVEL**
  - \([\text{assign}(\text{organ},\text{recipient})], \text{done}(\text{ensure}_{\text{appropriateness}}(\text{organ},\text{recipient}))\)

- \(\text{ensure}_{\text{appropriateness}}(o,r)\)
  - \(\text{assign}(o,r)\)
Context as source of interpretation

**ABSTRACT LEVEL**

- $O_{\text{ONT}}(\text{appropriate}(\text{distribution}))$
- $O_{\text{ONT}}(\text{ensure}_{\text{appropriateness}}(\text{organ}, \text{recipient}) < \text{do}(\text{assign}(\text{organ}, \text{recipient})))$
- $O_{\text{CARREL}}(\text{ensure}_{\text{quality}}(\text{organ}) < \text{do}(\text{assign}(\text{organ}, \text{recipient})))$
- $O_{\text{CARREL}}(\text{ensure}_{\text{compatibility}}(\text{organ}, \text{recipient}) < \text{do}(\text{assign}(\text{organ}, \text{recipient})))$

**Spanish National Health System**

**CONCRETE LEVEL**

- $\begin{array}{ll}
\text{assign}(\text{organ}, \text{recipient})
\end{array}$

**PROCEDURE LEVEL**

- $\begin{array}{ll}
\text{ensure}_{\text{appropriateness}}(o, r)
\end{array}$
- $\begin{array}{ll}
\text{ensure}_{\text{quality}}
\end{array}$
- $\begin{array}{ll}
\text{ensure}_{\text{compatibility}}
\end{array}$
- $\begin{array}{ll}
\text{assign}(o, r)
\end{array}$

Current version of the idea

**ABSTRACT LEVEL**

- $O_{\text{OB}}(\text{ONT, appropriate}(\text{distribution}))$
- $O_{\text{OB}}(\text{ONT, ensure}_{\text{appropriateness}}(\text{organ, recipient}) < \text{do}(\text{assign}(\text{organ, recipient})))$
- $O_{\text{OB}}(\text{ONT, ensure}_{\text{quality}}(\text{organ}) \text{ BEFORE do}(\text{assign}(\text{organ, recipient})))$

**Spanish National Health System**

**CONCRETE LEVEL**

- $\begin{array}{ll}
\text{OBLIGED}(\text{utter} (S7, W3, \text{quality ensuring}(\text{organ}))) \text{ IF (uttered}(S7, W3, \text{assign}(\text{organ, recipient})))
\end{array}$

**PROCEDURE LEVEL**

- $\begin{array}{ll}
\text{uttered}(S7, W3, \text{assign}(\text{organ, recipient}) \not\text{ or not uttered} (S7, W3, \text{quality ensuring}(\text{organ})) \rightarrow
\end{array}$

**AMELI implementation**
Problem 2: Defeasibility in human law

- Defeasibility = one or more norms defeated by addition of norms
- 2 levels:
  - Defeasibility of classification
    - semantics of concepts in norms extended/reduced/altered
  - Defeasibility of norms
    - impact & applicability of norm altered

Article 13

A13.1 OBLIGED((system DO record(procurement., sysLogs))
IF NOT(origin(procurement., decree(Minister.Of.Justice))))

A13.5 NOT(OBLIGED((system DO record(procurement., sysLogs))
IF (origin(procurement., linkage,) AND reported(linkage, sysLogs))))

Problem 2: Defeasibility in human law

- Option 1: Defeasibility handling in reasoning mechanism
  - there is no efficient implementation of defeasible logics!
- Option 2: “by-pass” defeasible reasoning
  - changes in law almost never occur

A13.1.5 OBLIGED((system DO record(procurement., sysLogs))
IF (NOT(origin(procurement., decree(Minister.Of.Justice)))
AND NOT(origin(procurement., linkage,)
AND reported(linkage, sysLogs))))

- changes occur often/periodically

A13.1 OBLIGED((system DO record(procurement., sysLogs))
IF (NOT(origin(procurement., decree(Minister.Of.Justice)))
AND NOT(CONDITIONAL.EXCEPTION(A13.1))))

A13.5 CONDITIONAL.EXCEPTION(A13.1)
IF (origin(procurement., linkage,)
AND reported(linkage, sysLogs))
Ongoing work: 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

IMPLEMENTATION ISSUES
Implementing Norms in eInstitutions

- Implementation of norms ≠ 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

Problem 3: Verifiability of norms

- Computational verifiable
  - Directly verifiable
  - Verifiable by the introduction of extra resources

- Non-computational verifiable

- Non-verifiable
  - Observable, but not decidable
  - Indirectly observable
  - Not verifiable at all
Safety and Soundness

- The concept of Norms 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

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

- Variant of Deontic Logic

\[
\text{OBLIGED, PERMITTED, FORBIDDEN} \\
\text{IF } C \\
\text{BEFORE } D, \text{ AFTER } D
\]
Representing Norms (II)

- examples:

  \[
  \text{FORBIDDEN}(\text{recipient}, (\text{in\_waiting\_list}(\text{hospital}_1) \land \text{in\_waiting\_list}(\text{hospital}_2) \land (\text{hospital}_1 \neq \text{hospital}_2)))
  \]

  \[
  \text{FORBIDDEN}(\text{person DO sell(organ)})
  \]

  \[
  \text{FORBIDDEN}((\text{allocator DO assign(organ, recipient)}) \land \text{NOT} (\text{hospital DONE ensure\_quality(organ)}))
  \]

  \[
  \text{OBLIGED}((\text{allocator DO assign(heart, recipient)}) \land \text{BEFORE} (\text{time(done(extraction(heart, donor))) + 6 hours}))
  \]

  \[
  \text{OBLIGED}(\text{ONT ENFORCE}(\text{FORBIDDEN}(\text{person DO sell(organ})))))
  \]

Implementing Norms in eInstitutions (II)

- Norms describe which states/actions within the e-institution should ideally take place
- Norms are too abstract, not operational
  - A norm implementation should be composed of:

  ```
  Norm FORBIDDEN(allocator DO assign(organ, recipient))
  condition done(assign(organ, recipient))
  detection detect_alarm(assign, starting)
  mechanism check(done(ensure_quality(organ)))
  sanction inform(board, "NOT(\text{done(ensure\_quality(organ))}\land \text{done(assign(organ, recipient))})")
  repair {stop_assignment(organ); record("\text{NOT(\text{done(ensure\_quality(organ))}\land \text{done(assign(organ, recipient))})", incident\_log); detect\_alarm(\text{ensure\_quality, done}); check(done(\text{ensure\_quality(organ)})); resume\_assignment(organ);}
  ```
Support for Implementing Norms (I)

- Norm enforcement is not centralized but distributed in a set of internal agents
  - They check if a given (observable) action was legal or illegal given the violation conditions defined for that context.
- The Agent Platform should assist the internal agents, providing fast, very efficient resources for norm enforcement as additional platform services and mechanisms.
  - A) Detection of the occurrence of an action
    - Internal agents may become overloaded checking ALL actions
    - black list mechanism (of actions to monitor) e.g., assign
    - action alarm mechanism (alarm to the internal agent)
    - Internal agent checks if conditions for a violation apply.

Support for Implementing Norms (II)

- B) Detection of activation/deactivation of norms
  - activation = when condition C is true
  - deactivation = when P holds, A is done or C is false
  - reaction time: time allowed between norm activation and reaction
  - Depending on the complexity to check C, the platform should implement the appropriate fast-access data structures and/or processing mechanisms to reduce computational burden
- C) Deadline control
  - a clock trigger mechanism to detect that a deadline has passed
Problem 4: need of tools

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