

# From Human Regulations to Regulated Software Agents' Behaviour.

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

Javier Vázquez-Salceda

May 20, 2005



Knowledge Engineering and Machine Learning Group  
UNIVERSITAT POLITÈCNICA DE CATALUNYA

<http://www.lsi.upc.es/~webia/KEMLG>



## MOTIVATION



Knowledge Engineering and Machine Learning Group  
UNIVERSITAT POLITÈCNICA DE CATALUNYA

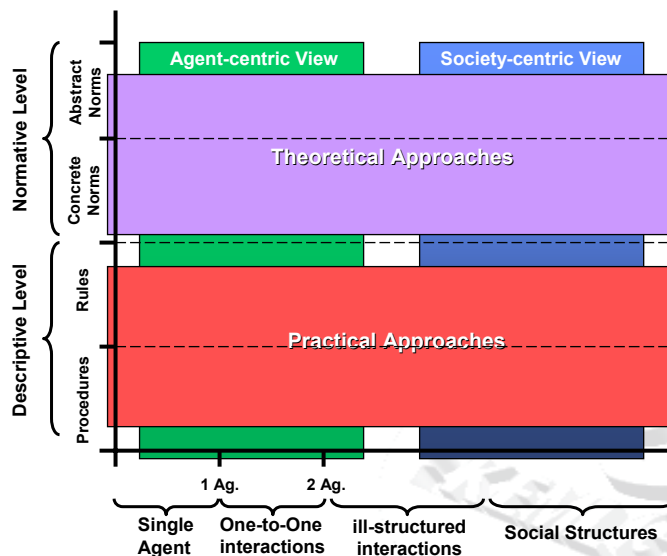
<http://www.lsi.upc.es/~webia/KEMLG>



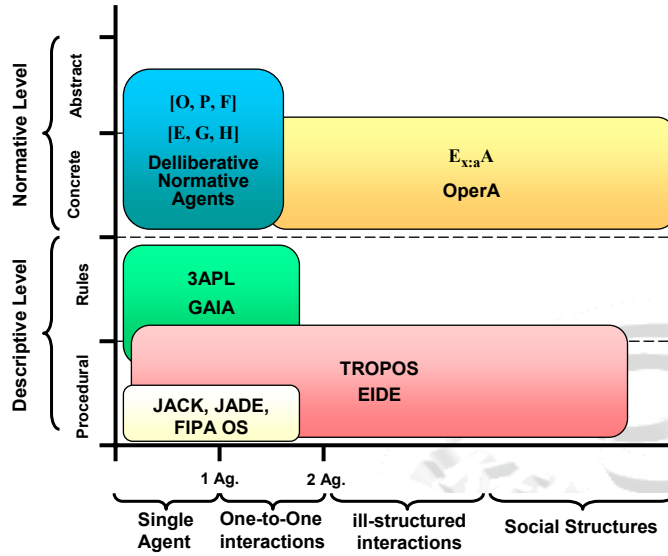
## eInstitutions and Norms

- 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)

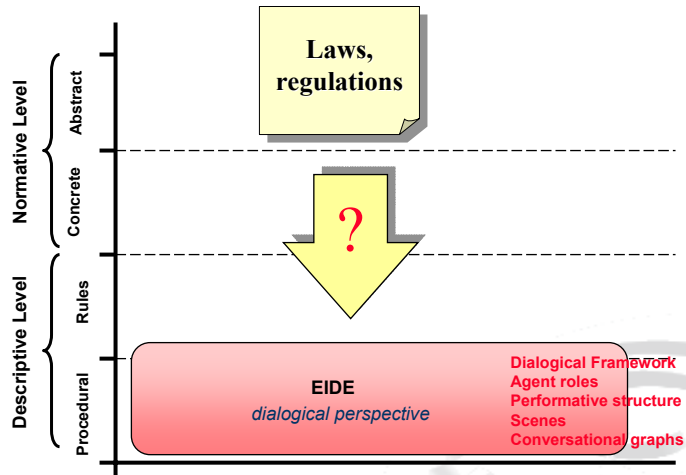


## Normative MAS: state of the Art (II)



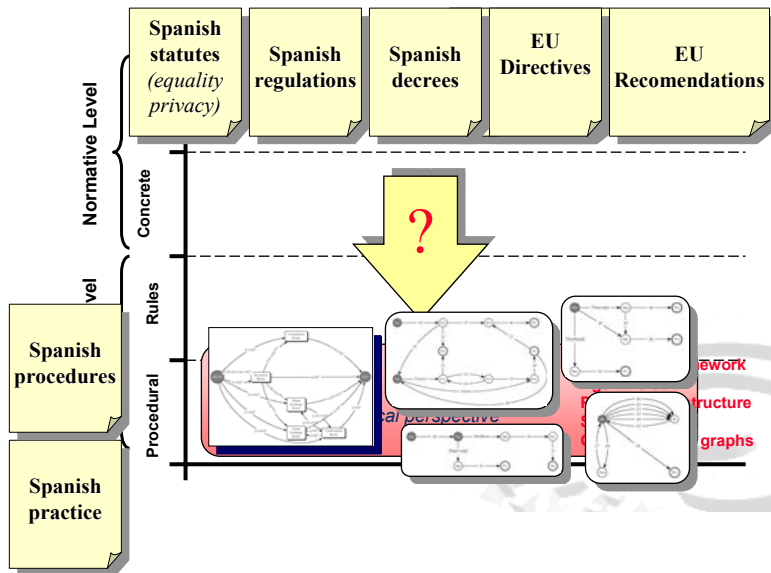
5

## Gap between Normative and Descriptive



6

## Example: Organ and Tissue Distribution



7

## Abstraction problem

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

### Example:

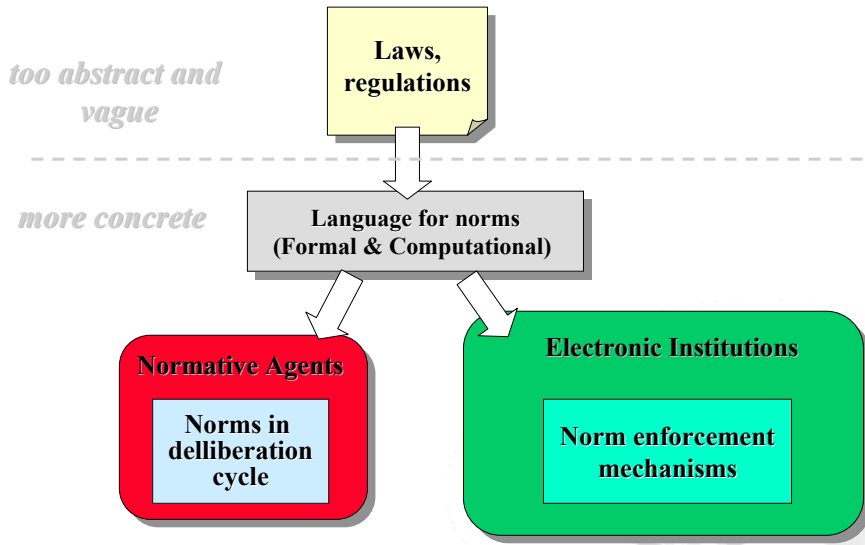
*Regulation: "It is forbidden to discriminate potential recipients of an organ based on their age (race, religion,...)"*

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

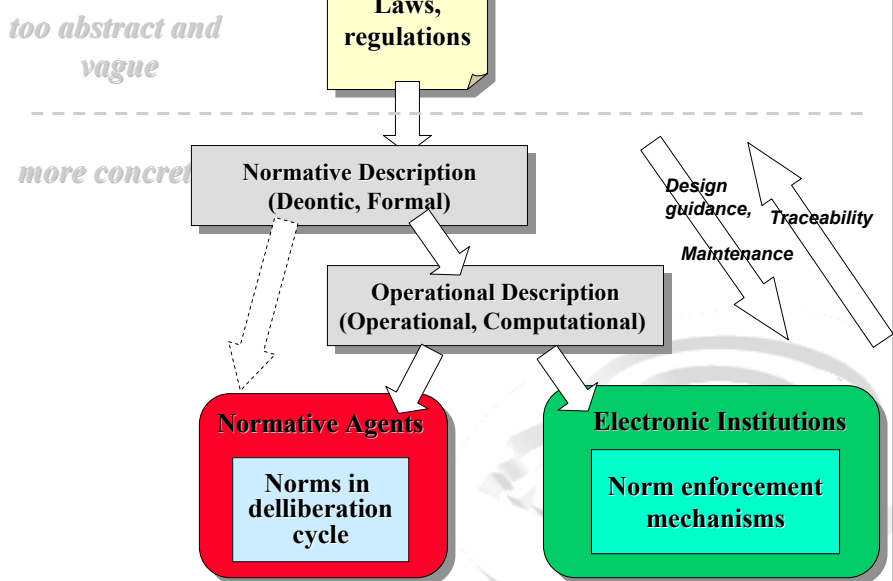
*Procedure: does not contain action "discriminate"*

8

### Filling the gap



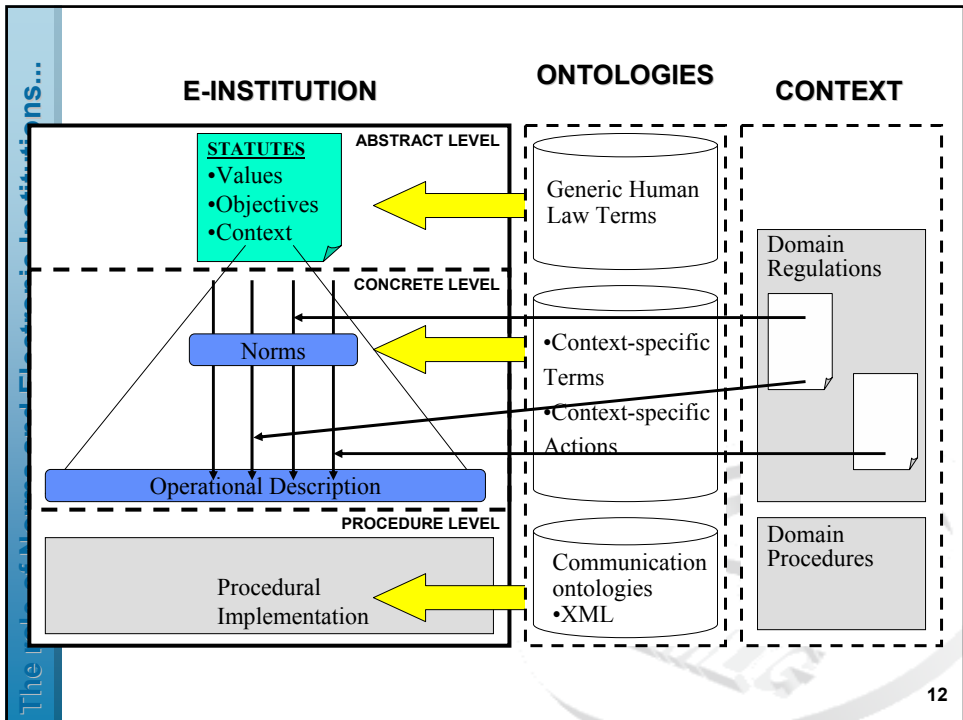
### Filling the gap



# From ABSTRACT to CONCRETE



Knowledge Engineering and Machine Learning Group  
 UNIVERSITAT POLITÈCNICA DE CATALUNYA  
<http://www.lsi.upc.es/~webia/KEMLG>



## 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

13

## Problem 1: Abstraction in Norms



- example 1: Abstract actions

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

$$\left. \begin{array}{l} \text{sign}(\text{donor}, \text{contract}) \cup \text{carry}(\text{donor}, \text{will}) \cup \\ \text{tell}(\text{donor}, \text{family}) \end{array} \right\} \Rightarrow_{\text{ONT}} \text{Consent}(\text{donor})$$

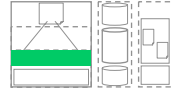
- example 2: Vague terms

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

$$\left. \begin{array}{l} O_{\text{ONT}}(\text{ensure\_quality}(\text{organ})) \wedge \\ O_{\text{ONT}}(\text{ensure\_compatibility}(\text{organ}, \text{recipient})) \end{array} \right\} \Rightarrow_{\text{ONT}} O_{\text{ONT}}(\text{appropriate}(\text{distribution}))$$

14

## From Normative to Operational



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

$$O_{\text{hosp}}(\text{consent}(\text{donor}(p,x)) < \text{do}(\text{transplant}(\text{hosp},x,p,q)))$$

↓

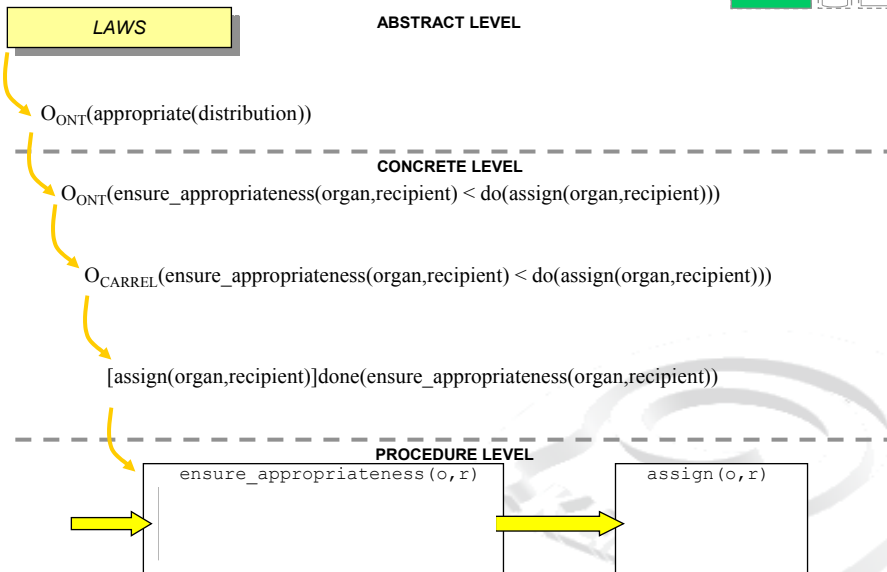
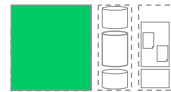
$$[\text{transplant}(\text{hosp},x,p,q)]\text{done}(\text{consent}(\text{donor}))$$

$$O_{\text{buyer}}(\text{pay}(\text{goods},\text{seller},\text{price}) < \text{do}(\text{exit}(\text{buyer})))$$

↓

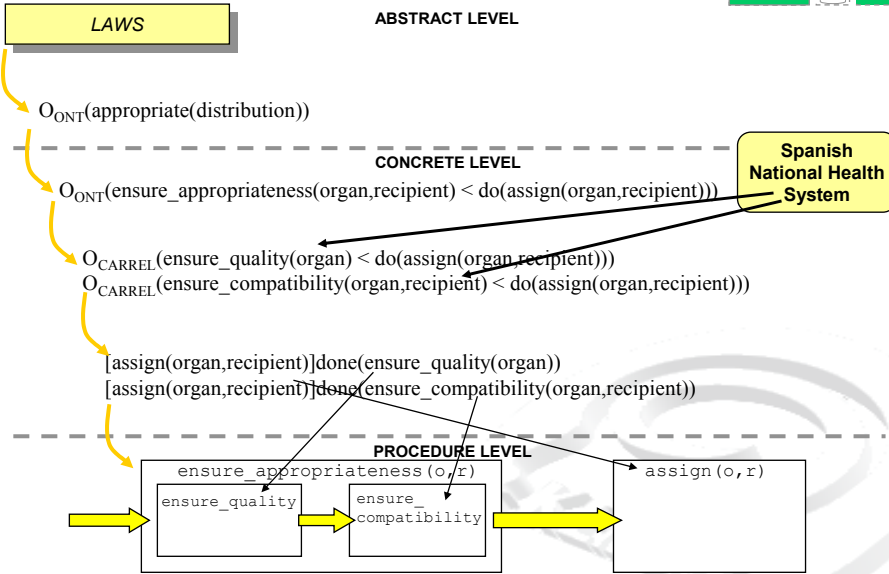
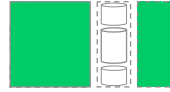
$$\text{not}(\text{done}(\text{pay}(\text{goods},\text{seller},\text{price}))) \rightarrow [\text{exit}(\text{buyer})]V(\text{fine}(\text{buyer}))$$

## Example

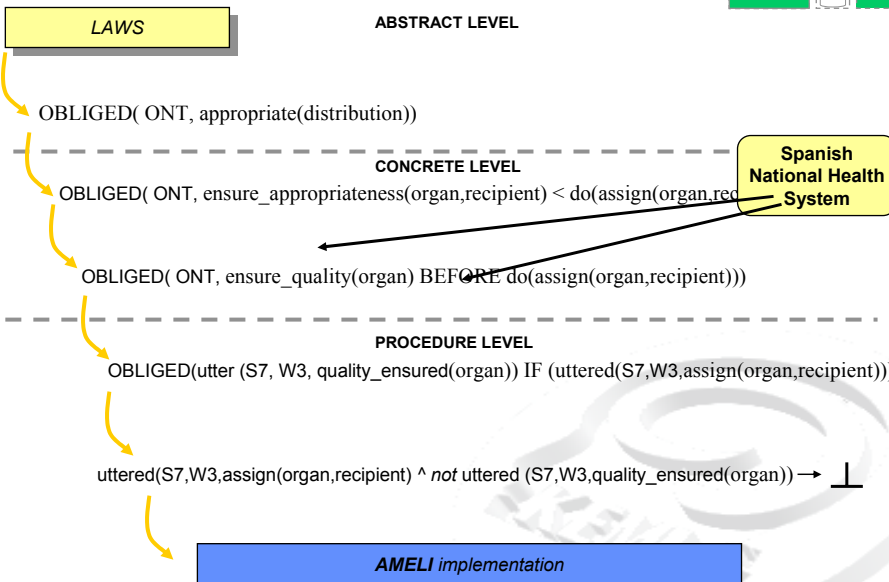
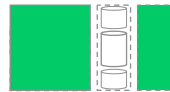




## Context as source of interpretation



## Current version of the idea



## 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<sub>i</sub>, sys\_logs))  
IF NOT(origin(procurement<sub>i</sub>, decree(Minister\_Of\_Justice))))

A13.5 NOT(OBLIGED((system DO record(procurement<sub>i</sub>, sys\_logs))  
IF (origin(procurement<sub>i</sub>, linkage<sub>j</sub>) AND reported(linkage<sub>j</sub>, sys\_logs))))

19

## 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<sub>i</sub>, sys\_logs))  
IF (NOT(origin(procurement<sub>i</sub>, decree(Minister\_Of\_Justice)))  
AND NOT(origin(procurement<sub>i</sub>, linkage<sub>j</sub>))  
AND reported(linkage<sub>j</sub>, sys\_logs))))

- changes occur often/periodically

A13.1 OBLIGED((system DO record(procurement<sub>i</sub>, sys\_logs))  
IF (NOT(origin(procurement<sub>i</sub>, decree(Minister\_Of\_Justice)))  
AND NOT(CONDITIONAL\_EXCEPTION(A13.1))))

A13.5 CONDITIONAL\_EXCEPTION(A13.1)  
IF (origin(procurement<sub>i</sub>, linkage<sub>j</sub>)  
AND reported(linkage<sub>j</sub>, sys\_logs))

20

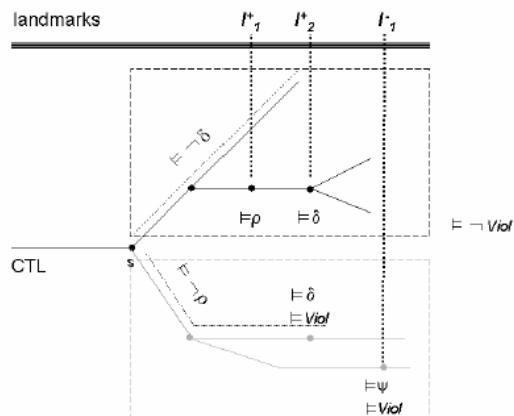
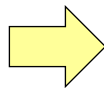
## Ongoing work: using landmarks for formal connection

- 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.

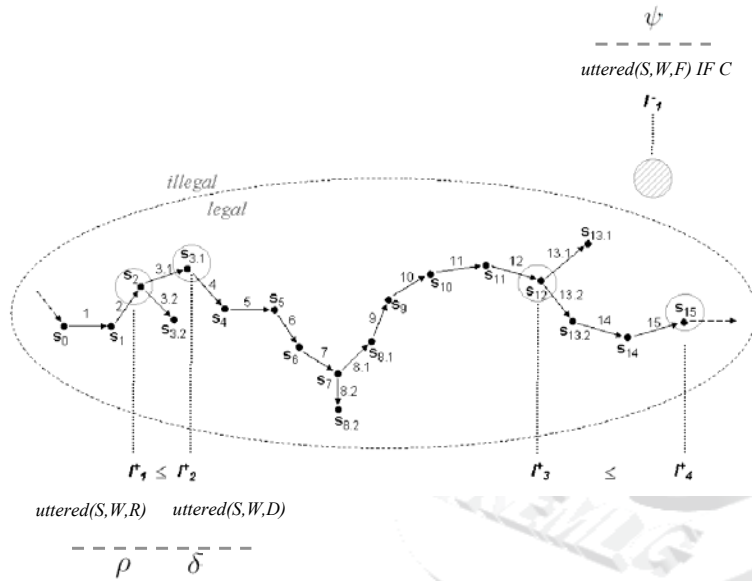
## From Norms to Landmark Patterns

$$O(\rho \leq \delta)$$

$$F\psi$$



## From Landmark Patterns to Protocols



## IMPLEMENTATION ISSUES



## Implementing Norms in Institutions

- Implementation of norms from institutional perspective  $\neq$  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

25

## 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

26

## 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

Safety

Soundness

## 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

**OBLIGED, PERMITTED, FORBIDDEN**  
**IF C**  
**BEFORE D, AFTER D**

→ variant of Deontic Logic

## Representing Norms (II)

- examples:

```
FORBIDDEN(recipient, (in_waiting_list(hospital1) ∧
in_waiting_list(hospital2) ∧ (hospital1 ≠ hospital2)))
```

```
FORBIDDEN(person DO sell(organ))
```

```
FORBIDDEN((allocator DO assign(organ, recipient))
IF NOT(hospital DONE ensure_quality(organ)))
```

```
OBLIGED((allocator DO assign(heart, recipient))
BEFORE (time(done(extraction(heart, donor))) + 6hours))
```

```
OBLIGED(ONT ENFORCE(FORBIDDEN(person DO sell(organ))))
```

29

## 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	IF NOT(hospital DONE ensure_quality(organ))
Violation condition	NOT(done(ensure_quality(organ))) AND done(assign(organ, recipient))
Detection mechanism	{detect_alarm(assign, 'starting'); check(done(ensure_quality(organ)))};
Sanction	inform(board, "NOT(done(ensure_quality(organ))) AND done(assign(organ, recipient))")
Repairs	{stop_assignment(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_assignment(organ);}

30

## 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.

31

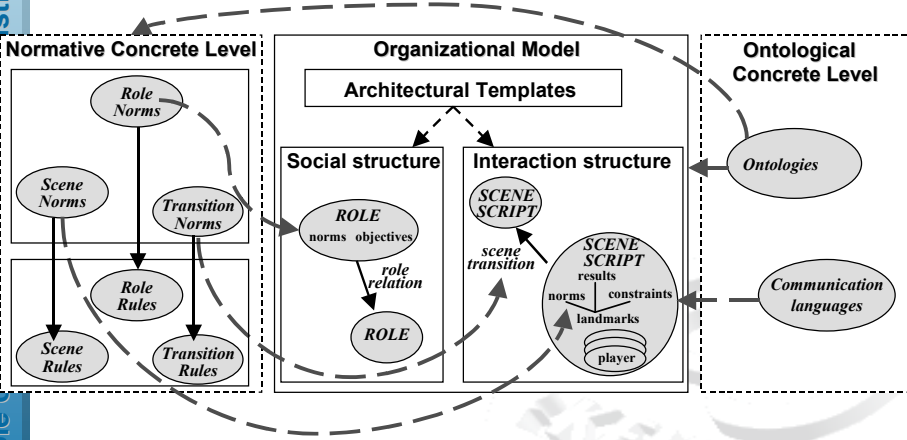
## Support for Implementing Norms (II)

- 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

32



### Problem 4: need of tools



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