Norms and Electronic Institutions for Behaviour Regulation in Distributed Systems.

Applications to eContracting Environments

Javier Vázquez-Salceda

April 1, 2008

Knowledge Engineering and Machine Learning Group
UNIVERSITAT POLITÈCNICA DE CATALUNYA

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

Contents

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

Introduction

Knowledge Engineering and Machine Learning Group

UNIVERSITAT POLITÈCNICA DE CATALUNYA

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

Introduction (I)

- Now a days, computing trends move toward **distributed solutions**
  - computer systems are networked into large distributed systems;
  - processing power can been 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 *collectivelly adapt* to unforeseen events
- Problem: how can we meet all these spectatives?

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 decissions and to adapt to new, unexpected situations.

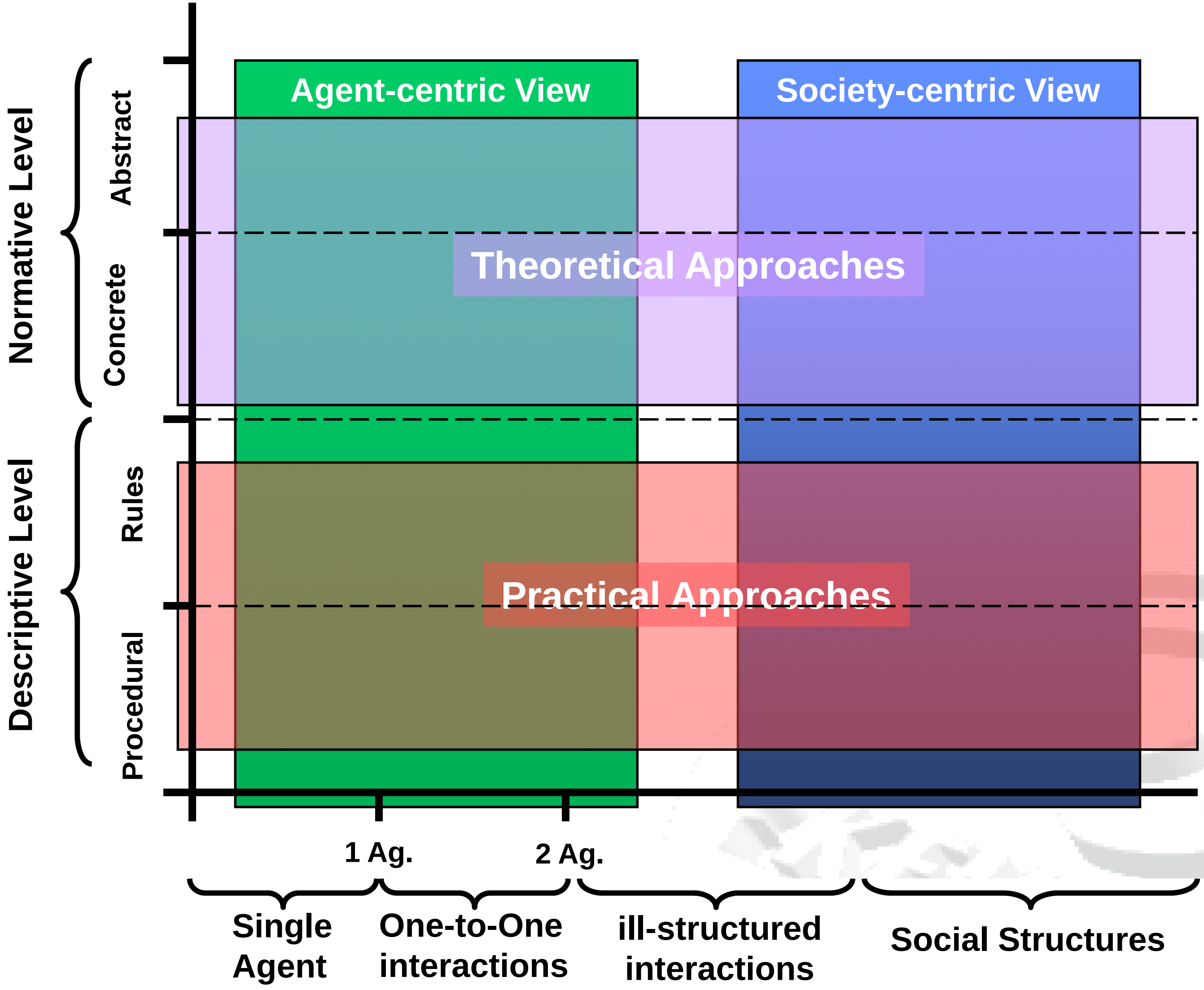
- We want agents to behave as expected, in order to achieve one or several goals. Therefore some **control** should be applied to the agents' behaviour.

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

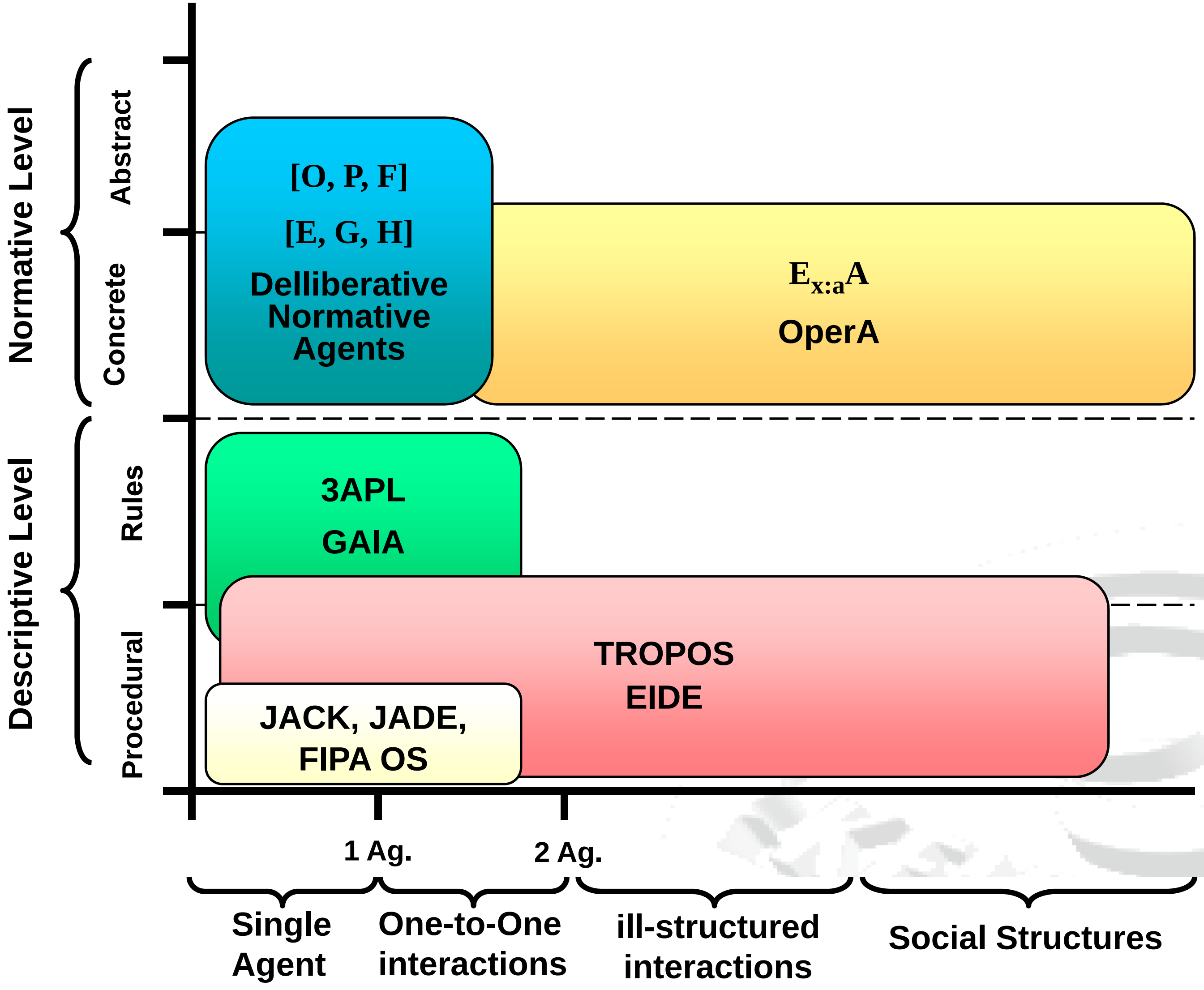
Introduction (IV)

- **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

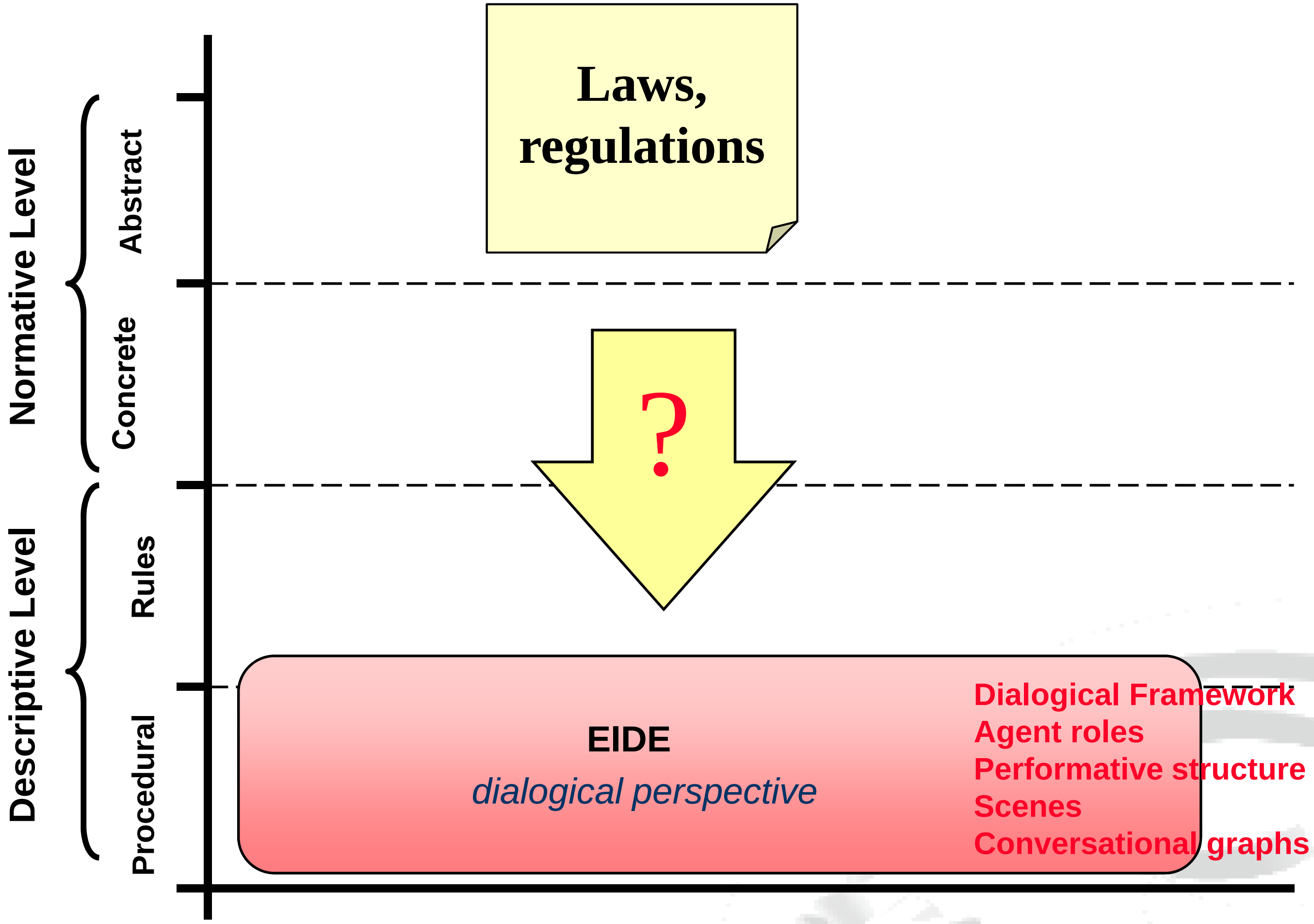
Normative MAS: state of the Art (I)



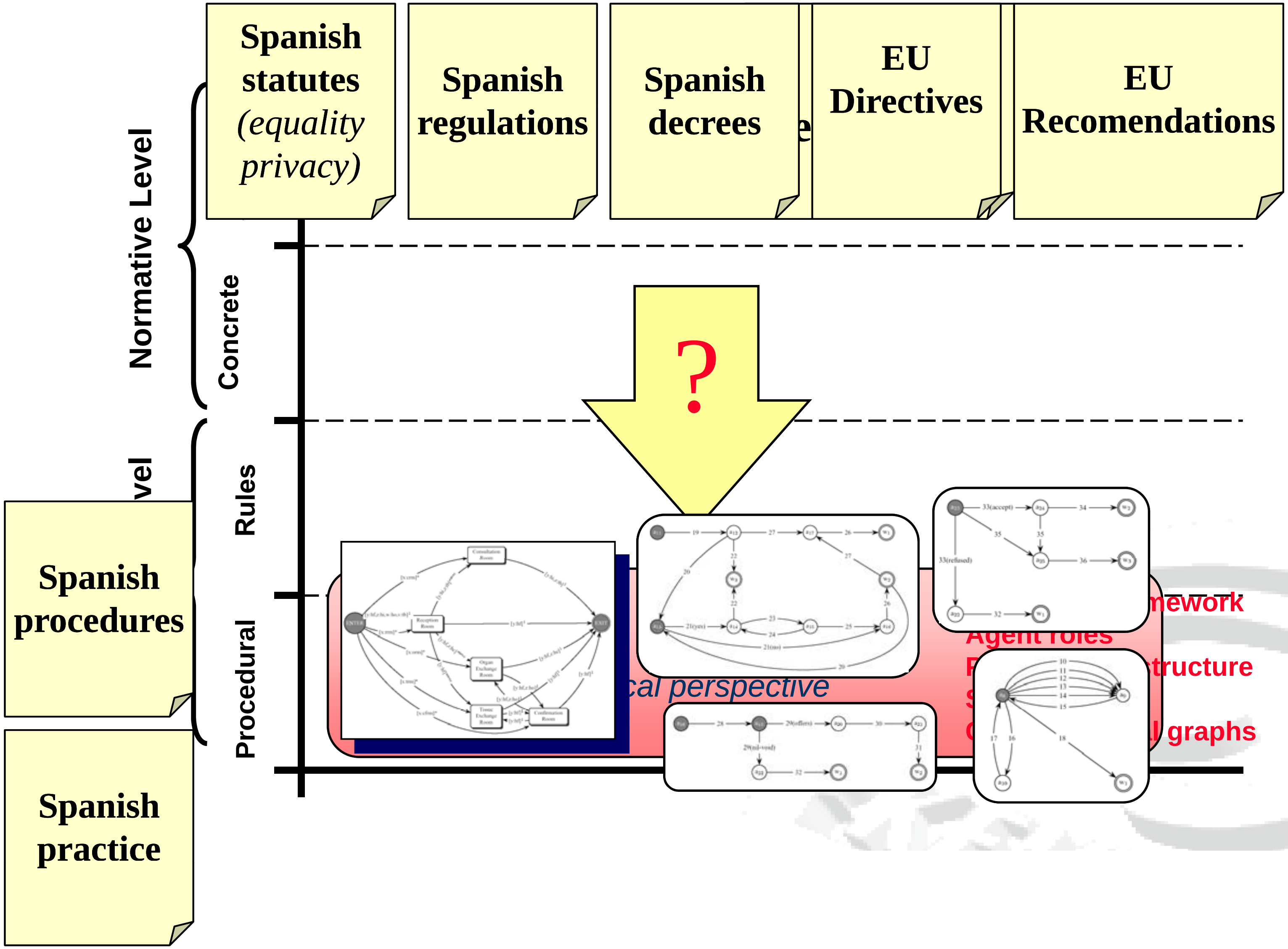
Normative MAS: state of the Art (II)



Gap between Normative and Descriptive



Example: Organ and Tissue Distribution



Spanish statutes *(equality privacy)*

Spanish regulations

Spanish decrees

EU Directives

EU Recomendations

Spanish procedures

Spanish practice

Abstraction problem

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

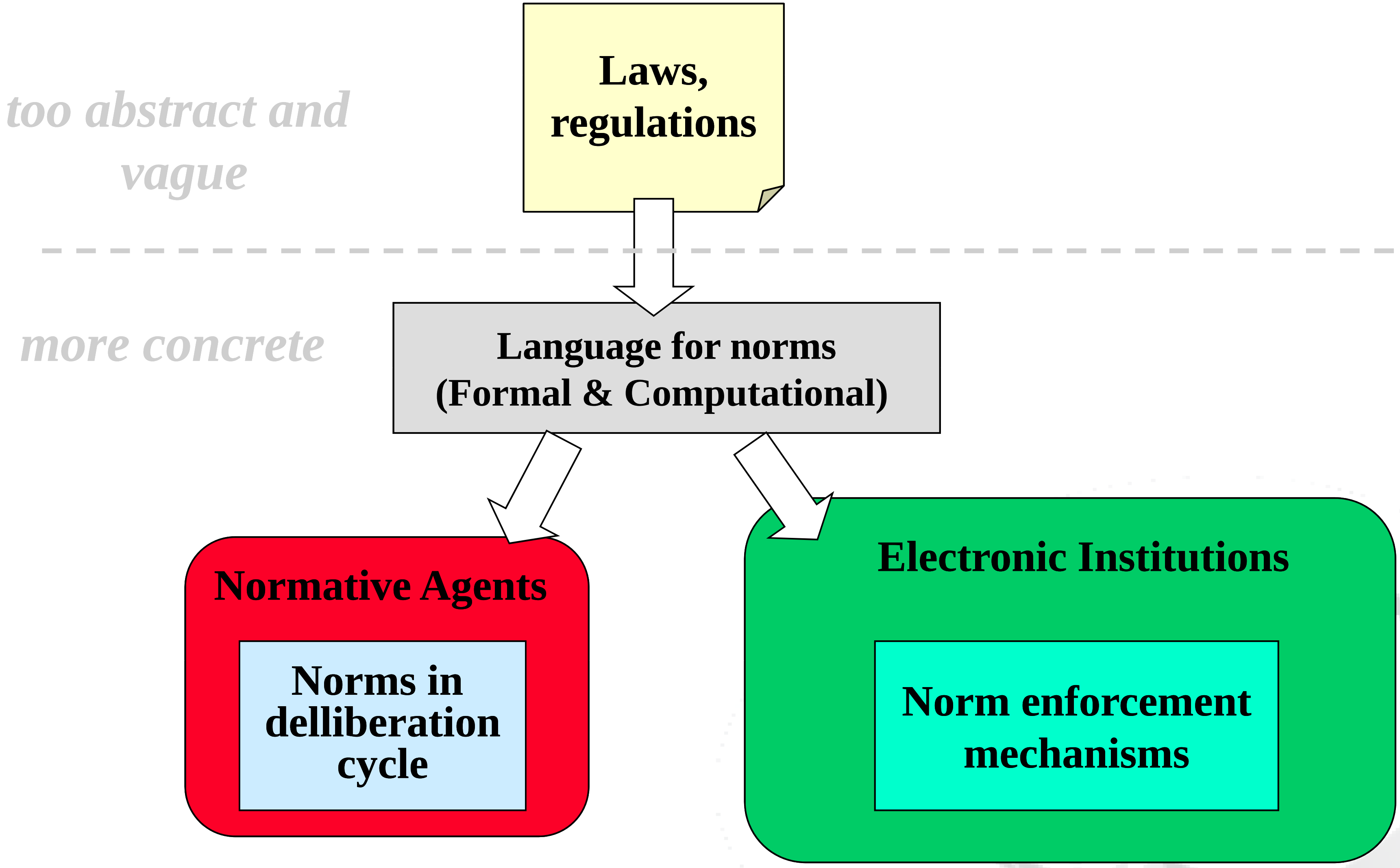
Example:

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

Formal norm: F(discriminate(x,y,age))

Procedure: does not contain action "discriminate"

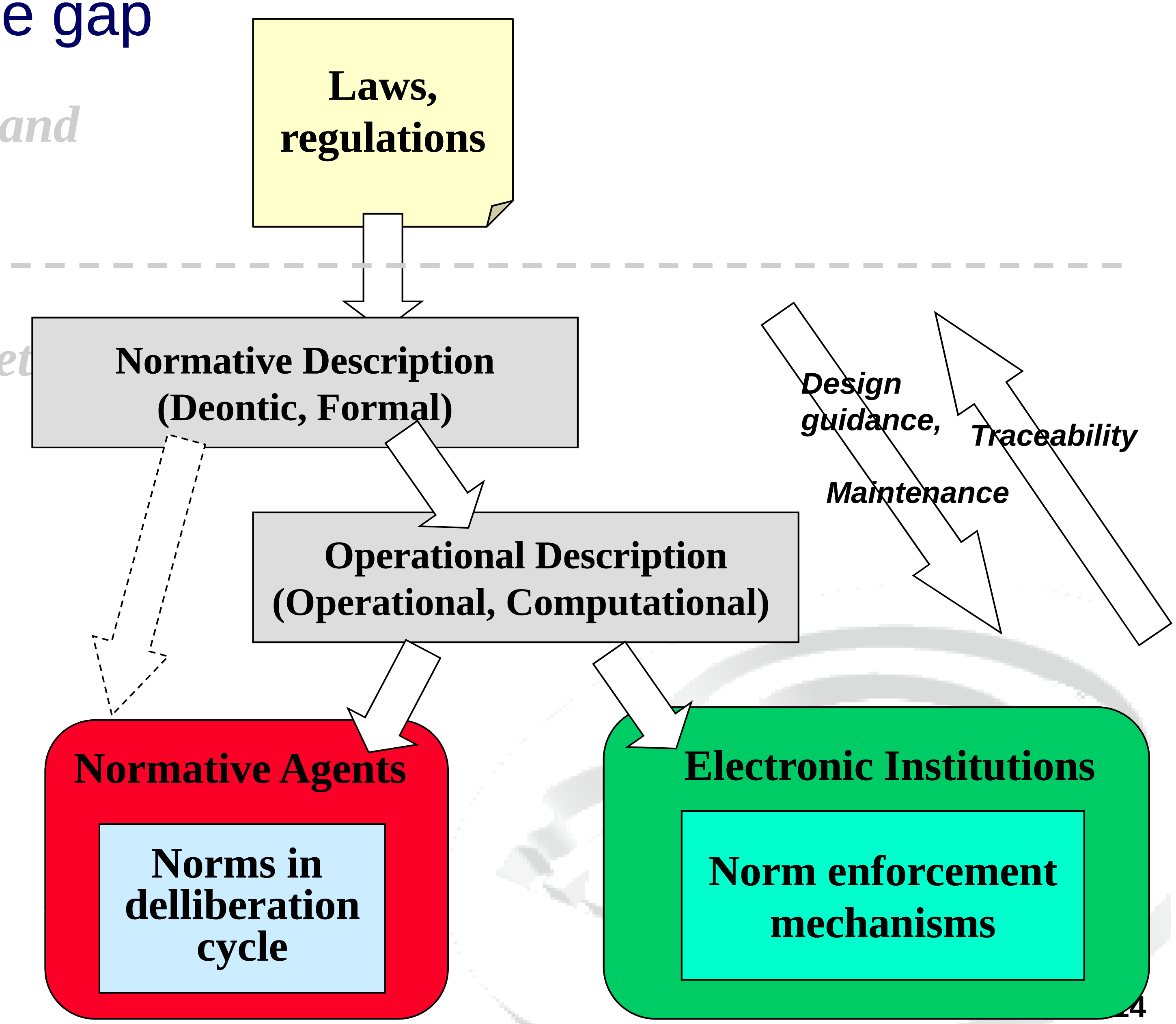
Filling the gap


too abstract and vague
Laws, regulations
more concrete
Language for norms (Formal & Computational)
Normative Agents
Norms in delliberation cycle
Electronic Institutions
Norm enforcement mechanisms

Filling the gap

too abstract and vague

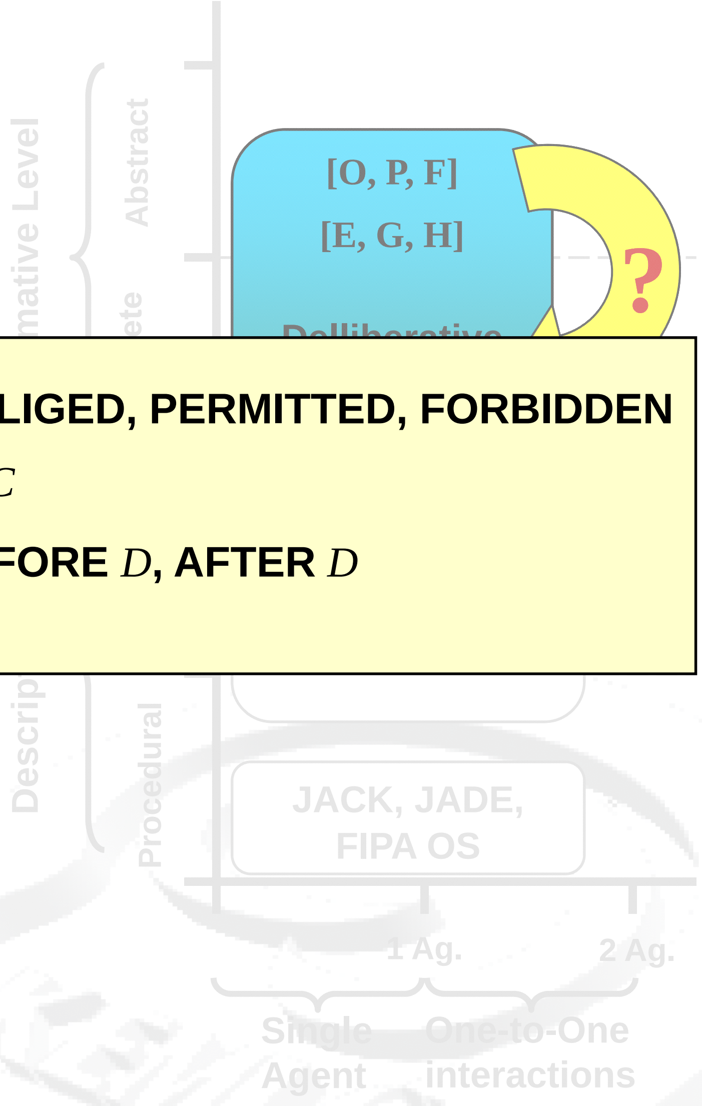
more concret


Laws, regulations
Normative Description (Deontic, Formal)
Operational Description (Operational, Computational)
Design guidance,
Maintenance
Traceability
Normative Agents
Norms in dellіberation 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 aspe
  - Norms are relativized to roles



OBLIGED, PERMITTED, FORBIDDEN

IF *C*

BEFORE *D*, **AFTER** *D*

variant of Deontic Logic

- The representation should be easily parseable and usable by agents

Representing Norms (II)

- Type 1: *Unconditional norms about predicates*
  - the norms on the value of *P* are active at all times:

$$\text{OBLIGED}(a, P) \qquad \text{PERMITTED}(a, P) \qquad \text{FORBIDDEN}(a, P)$$

  - an example:

$$\text{FORBIDDEN}(recipient, (in\_waiting\_list(hospital_1) \wedge in\_waiting\_list(hospital_2) \wedge (hospital_1 \neq hospital_2)))$$

- Type 2: *Unconditional norms about actions*
  - the norms on the execution of A are active at all times:

$$\text{PERMITTED}(a \text{ DO } A) \qquad \text{FORBIDDEN}(a \text{ DO } A)$$

  - an example:

$$\text{FORBIDDEN}(person \text{ DO } sell(organ))$$

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:

$$\begin{array}{ll} \text{OBLIGED}((a, P) \text{ IF } C) & \text{OBLIGED}((a \text{ DO } A) \text{ IF } C) \\ \text{PERMITTED}((a, P) \text{ IF } C) & \text{PERMITTED}((a \text{ DO } A) \text{ IF } C) \\ \text{FORBIDDEN}((a, P) \text{ IF } C) & \text{FORBIDDEN}((a \text{ DO } A) \text{ IF } C) \end{array}$$

  - an example:

$$\begin{array}{l} \text{FORBIDDEN}((allocator \text{ DO } assign(organ, recipient)) \\ \quad \text{IF NOT}(hospital \text{ DONE } ensure\_quality(organ))) \end{array}$$

Representing Norms (IV)

- Type 4: *Conditional norms with Deadlines*
  - the activation of norms is defined by a deadline

$$\text{OBLIGED}((a, P)\ \text{BEFORE}\ D)$$
$$\text{PERMITTED}((a\ \text{DO}\ A)\ \text{AFTER}\ D)$$
$$\text{FORBIDDEN}((a, P)\ \text{BEFORE}\ D)$$

  - absolute and relative deadlines:

  - 23:59:00 09/05/2004 $time(done(assign(organ, recipient))) + 5min$

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

Representing Norms (V)

- Type 5: *Obligations of enforcement of norms*
  - norms concerning agent b generate obligations on agent a:

$$\mathsf{OBLIGED}(a\ \mathsf{ENFORCE}(\mathsf{OBLIGED}(b...)))$$
$$\mathsf{OBLIGED}(a\ \mathsf{ENFORCE}(\mathsf{PERMITTED}(b...)))$$
$$\mathsf{OBLIGED}(a\ \mathsf{ENFORCE}(\mathsf{FORBIDDEN}(b...)))$$

  - an example:

$$\mathsf{OBLIGED}(ONT\ \mathsf{ENFORCE}(\mathsf{FORBIDDEN}(person\ \mathsf{DO}\ sell(organ))))$$

Norms and Agents

Knowledge Engineering and Machine Learning Group
UNIVERSITAT POLITÈCNICA DE CATALUNYA

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

Norms and eInstitutions for Behaviour Regulation

Normative Agents (I)

Ensuring proper agent behaviour with norms

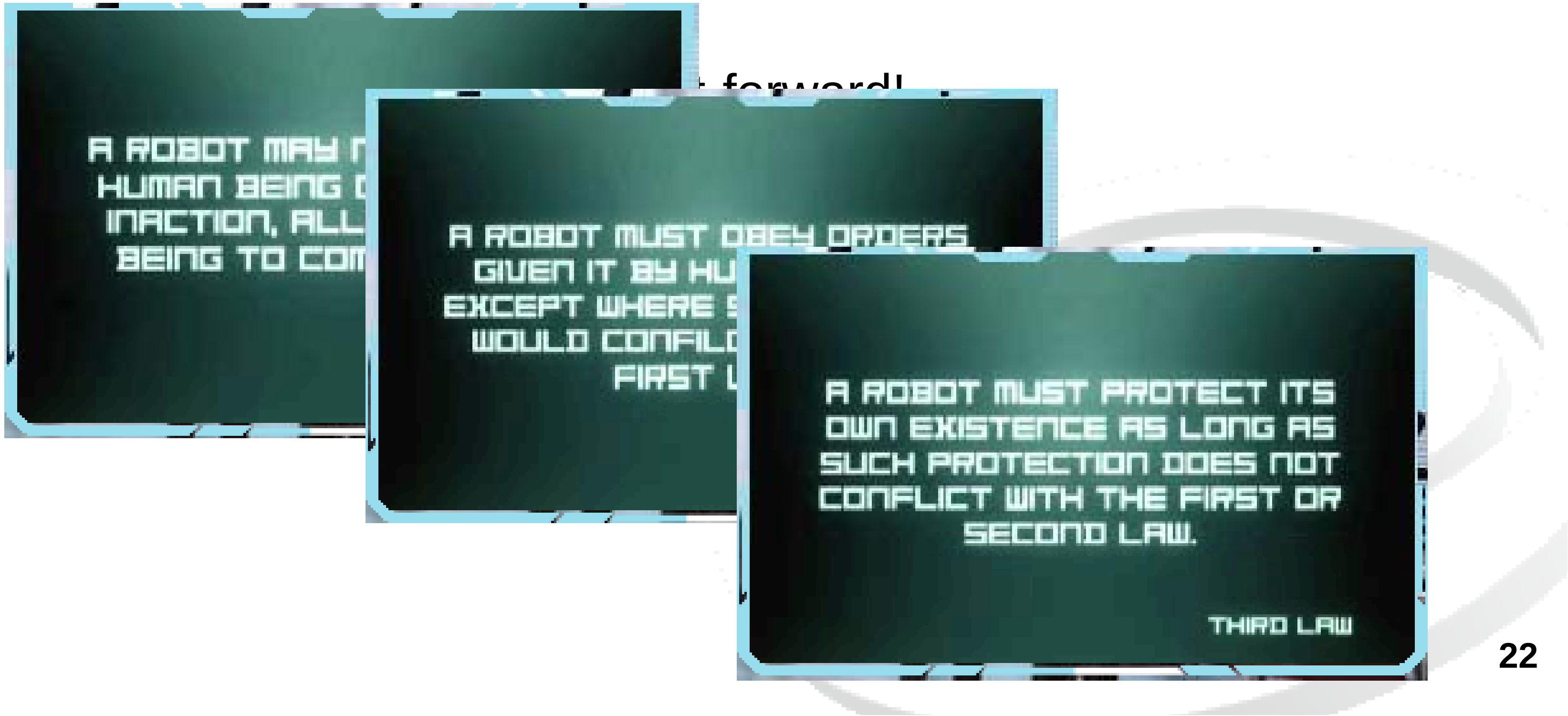
- Medicine is a very sensible domain
  - We mush ensure proper behaviour of agents
  - Agents should keep a certain autonomy

Agents

Autonomy VS **Control**

- We can express agents´ acceptable behaviour with norms

t forward!



A ROBOT MAY N
HUMAN BEING O
INACTION, ALL
BEING TO COM

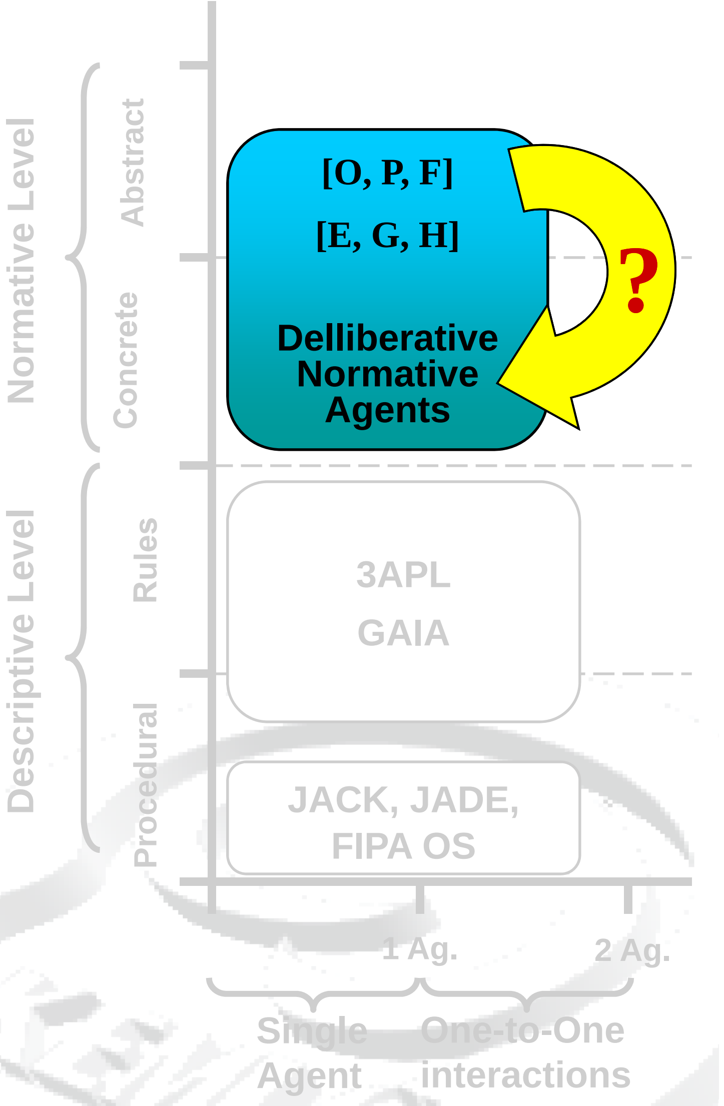
A ROBOT MUST OBEY ORDERS
GIVEN IT BY HU
EXCEPT WHERE S
WOULD CONFLIC
FIRST L

A ROBOT MUST PROTECT ITS OWN EXISTENCE AS LONG AS SUCH PROTECTION DOES NOT CONFLICT WITH THE FIRST OR SECOND LAW.

THIRD LAW

Normative Agents (II)

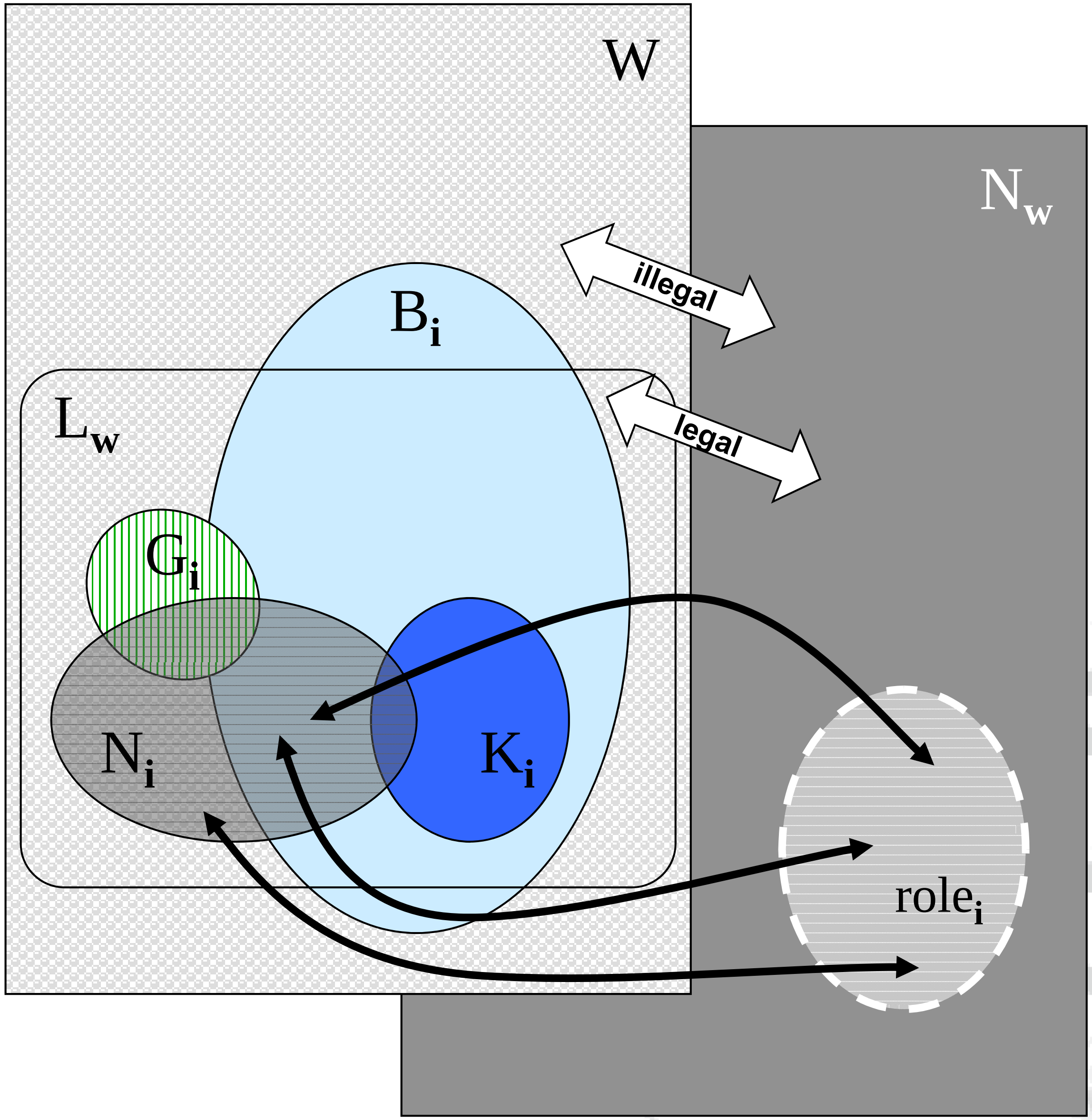
- Problem 1: Which is the relation between the norms and the agents beliefs, desires and intentions?
- Problem 2: How exactly can norms define acceptable behaviour?
- Idea: We should first analyse the impact of norms on cognitive agents from a theoretical perspective.



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)



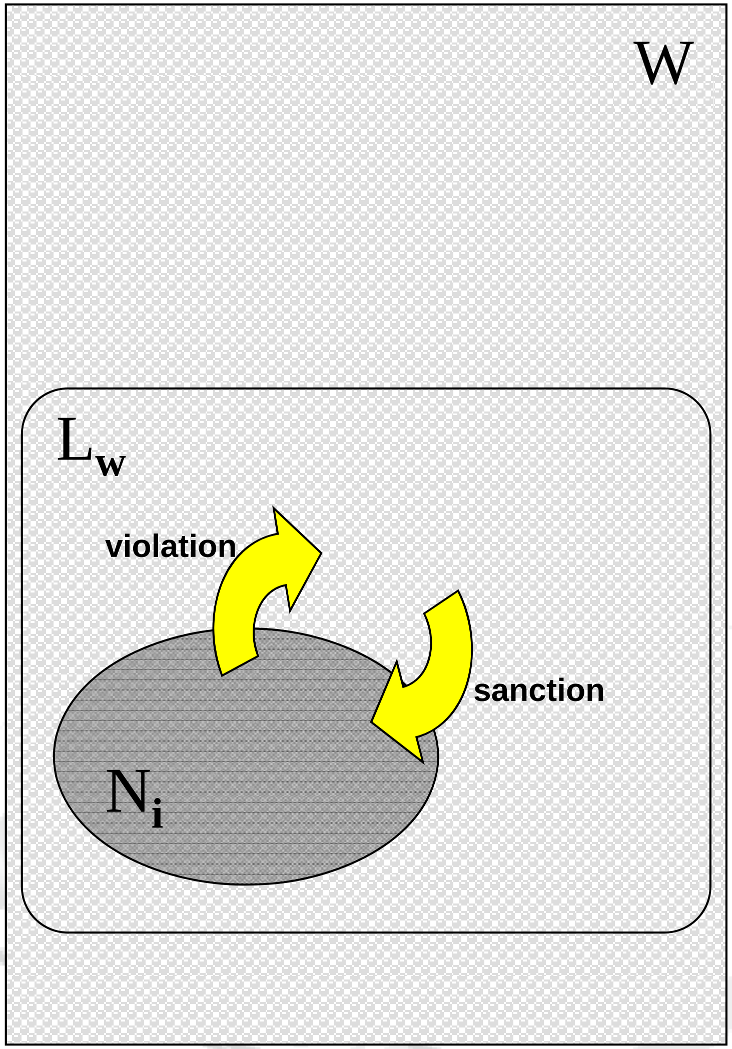
Normative Agents (V)

Safety and Soundness

- The concept of legally accessible worlds allows to describe
  - wanted (legal) and unwanted (illegal) behaviour
  - acceptable (safe) and unnacceptable (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

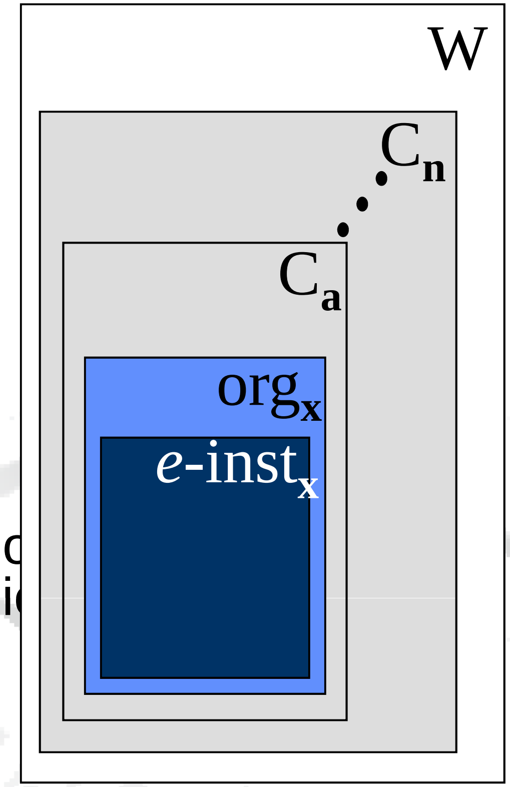
Soundness



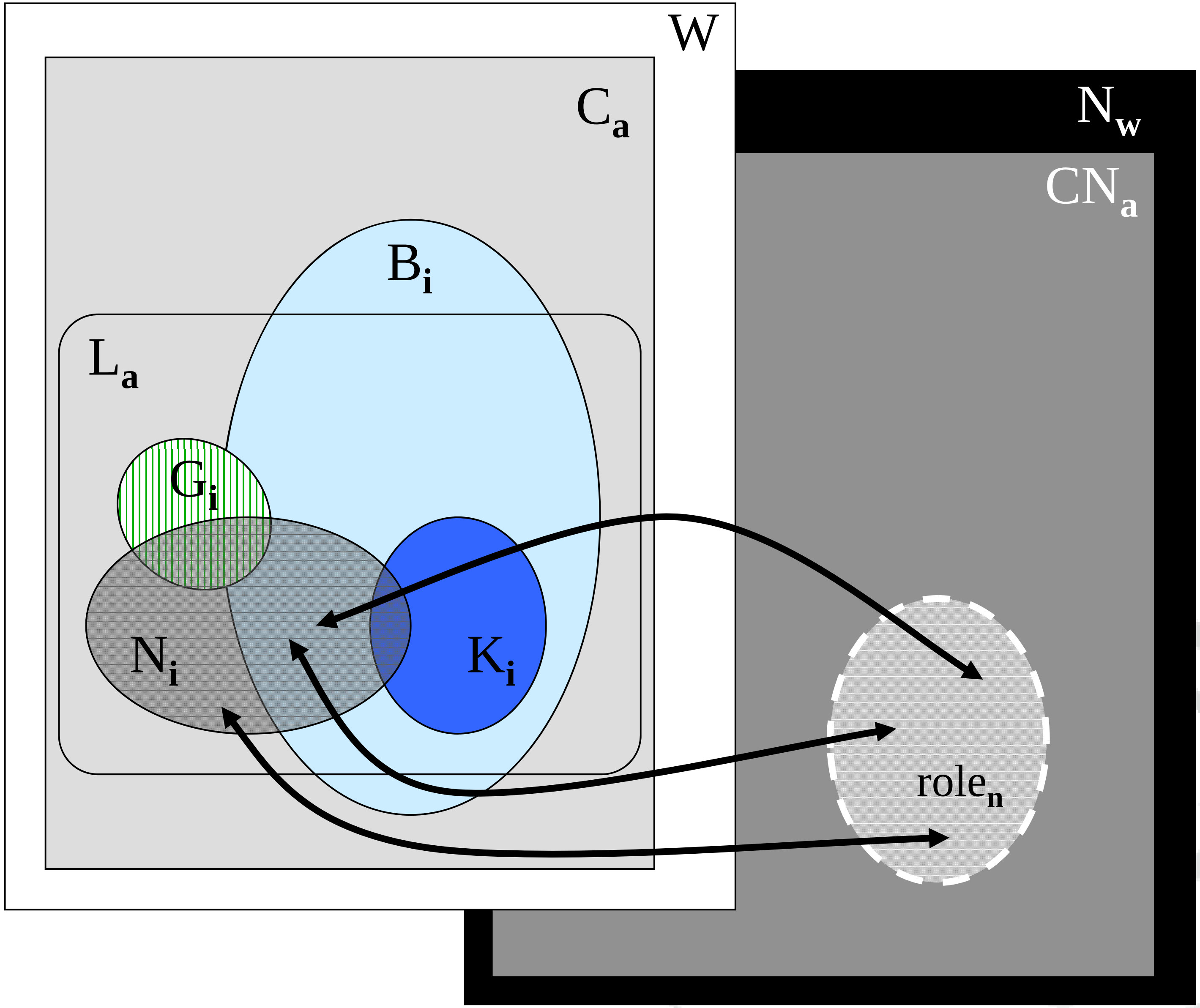
Normative Agents (VI)

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 a 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 representatio for all contexts, but interconnected ontologi (multi-contextual ontologies).

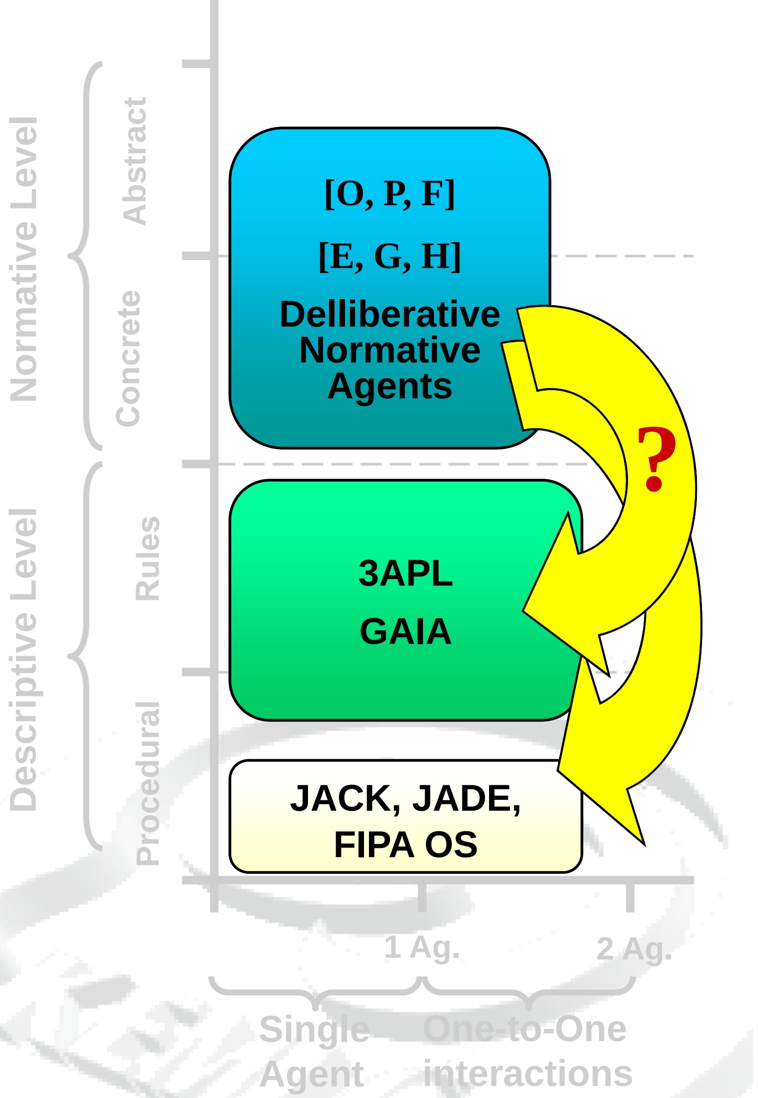


Normative Agents (VII)



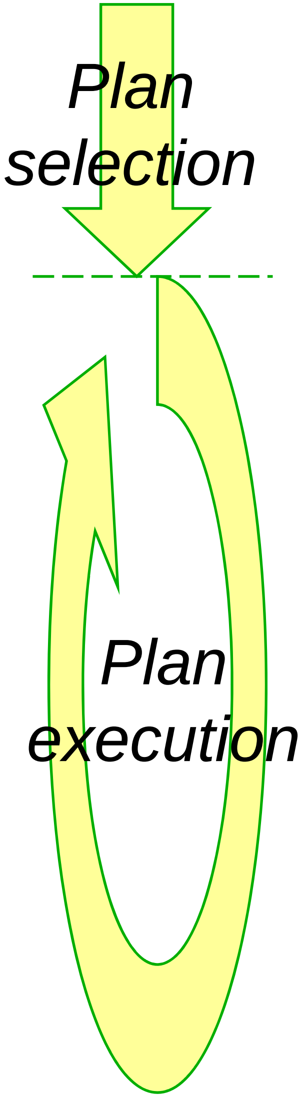
Implementing Normative Agents (I)

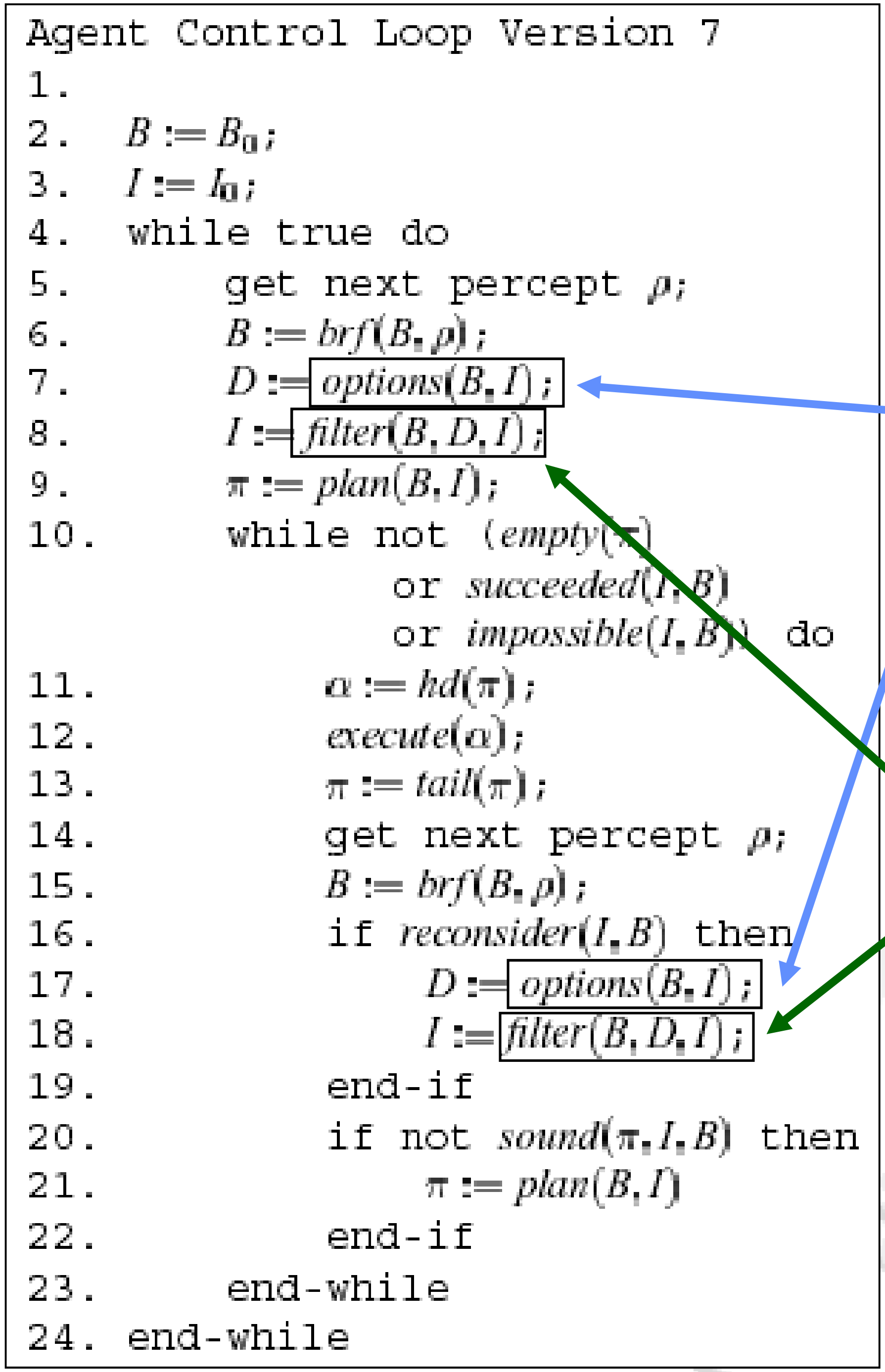
- Problem: HOW to introduce norms in the existing agent implementations?
- There are already implementations based in the BDI agent framework
  - E.g., 3APL agents , JACK agents, JADEx agents.
- Idea: Extend the BDI interpreter to include norms.



Norms and eInstitutions for Behaviour Regulation

Norms and Agents (IX)





```
Agent Control Loop Version 7

    $B := B_0$;
    $I := I_0$;
    while true do
        get next percept $\rho$;
        $B := brf(B, \rho)$;
        $D := \boxed{options(B, I);}$
        $I := \boxed{filter(B, D, I);}$
        $\pi := plan(B, I)$;
        while not $(empty(\pi)$
                  or $succeeded(I, B)$
                  or $impossible(I, B))$ do
            $\alpha := hd(\pi)$;
            $execute(\alpha)$;
            $\pi := tail(\pi)$;
            get next percept $\rho$;
            $B := brf(B, \rho)$;
            if $reconsider(I, B)$ then
                $D := \boxed{options(B, I);}$
                $I := \boxed{filter(B, D, I);}$
            end-if
            if not $sound(\pi, I, B)$ then
                $\pi := plan(B, I)$
            end-if
        end-while
end-while
```

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

Norms in Agent Platforms:

Electronic Institutions

Knowledge Engineering and Machine Learning Group
UNIVERSITAT POLITÈCNICA DE CATALUNYA

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

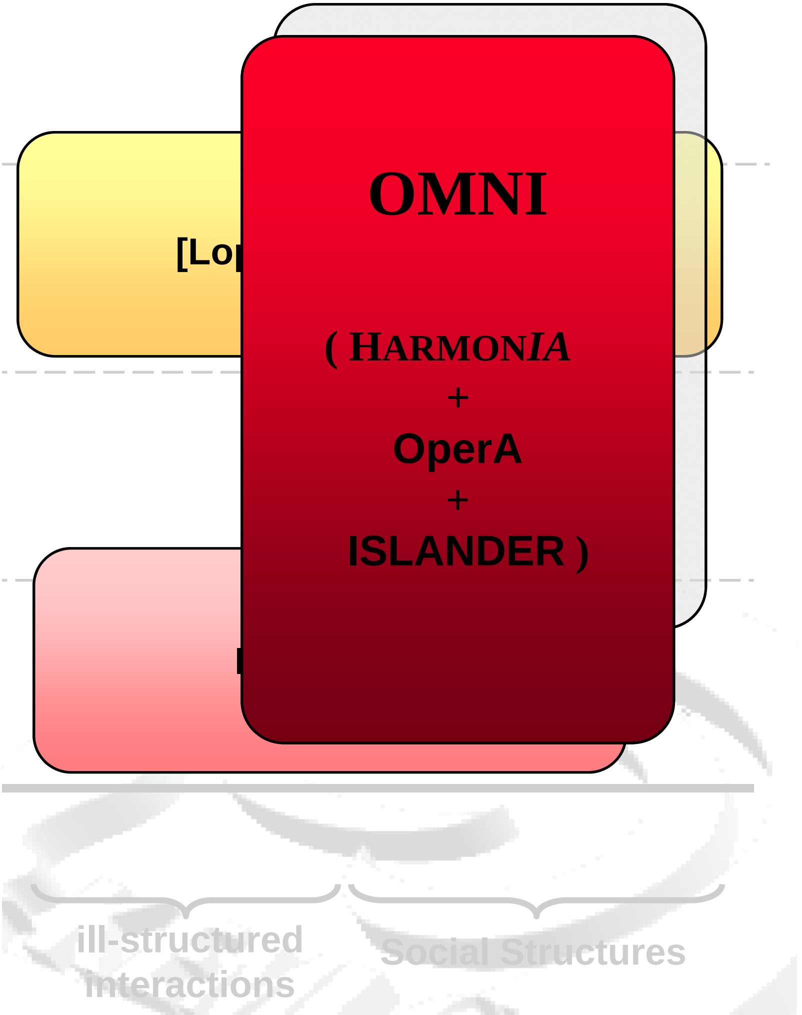
Electronic Institutions (I)

- 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).

  - Agent **behaviour guided by Norms**

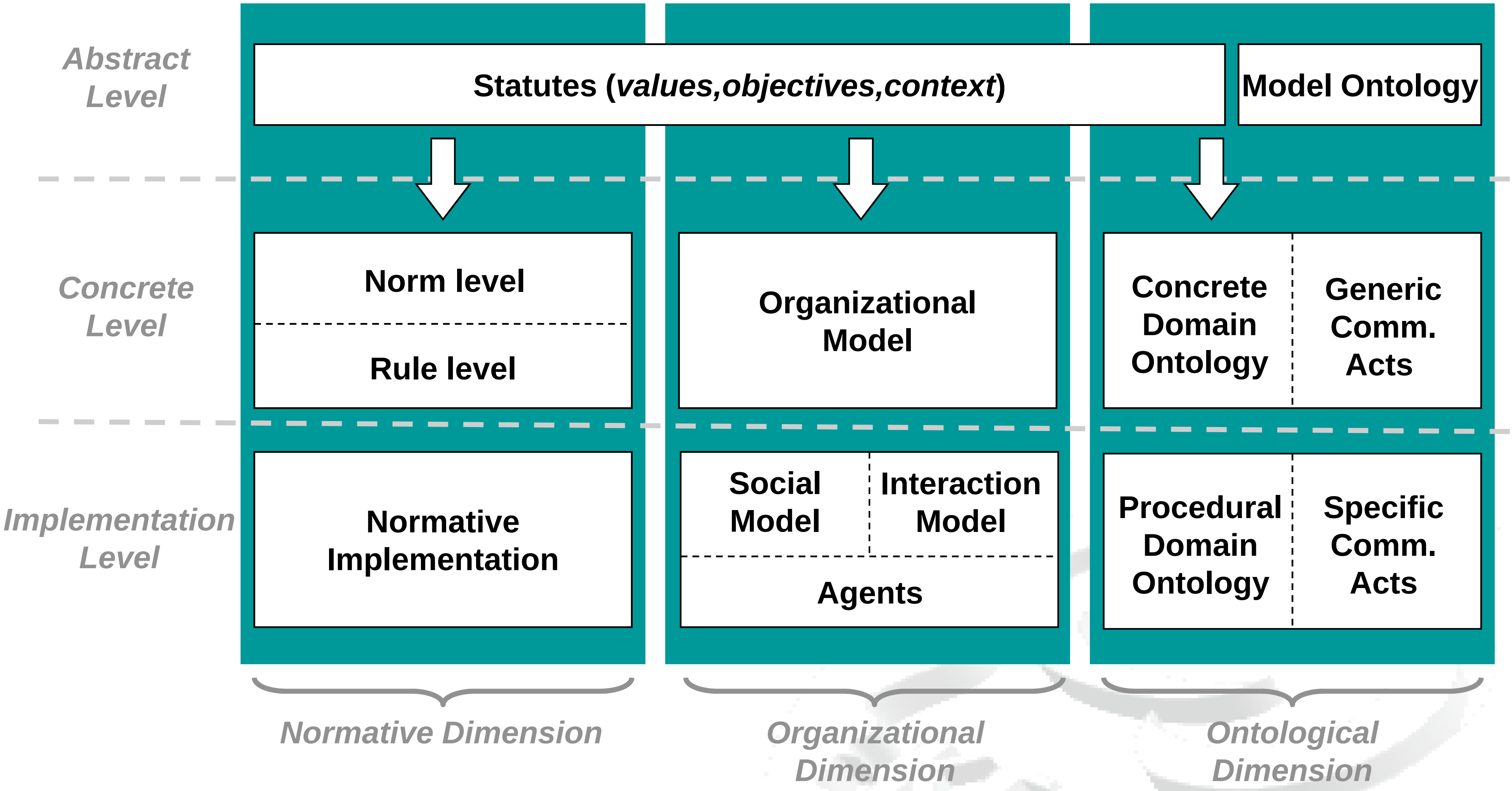
Electronic Institutions (II)

- Problem: no connection between theoretical work on eInstitutions and practical implementations on eInstitutions
- First proposal: the **HARMONIA** framework
- Ongoing work: the **OMNI** framework



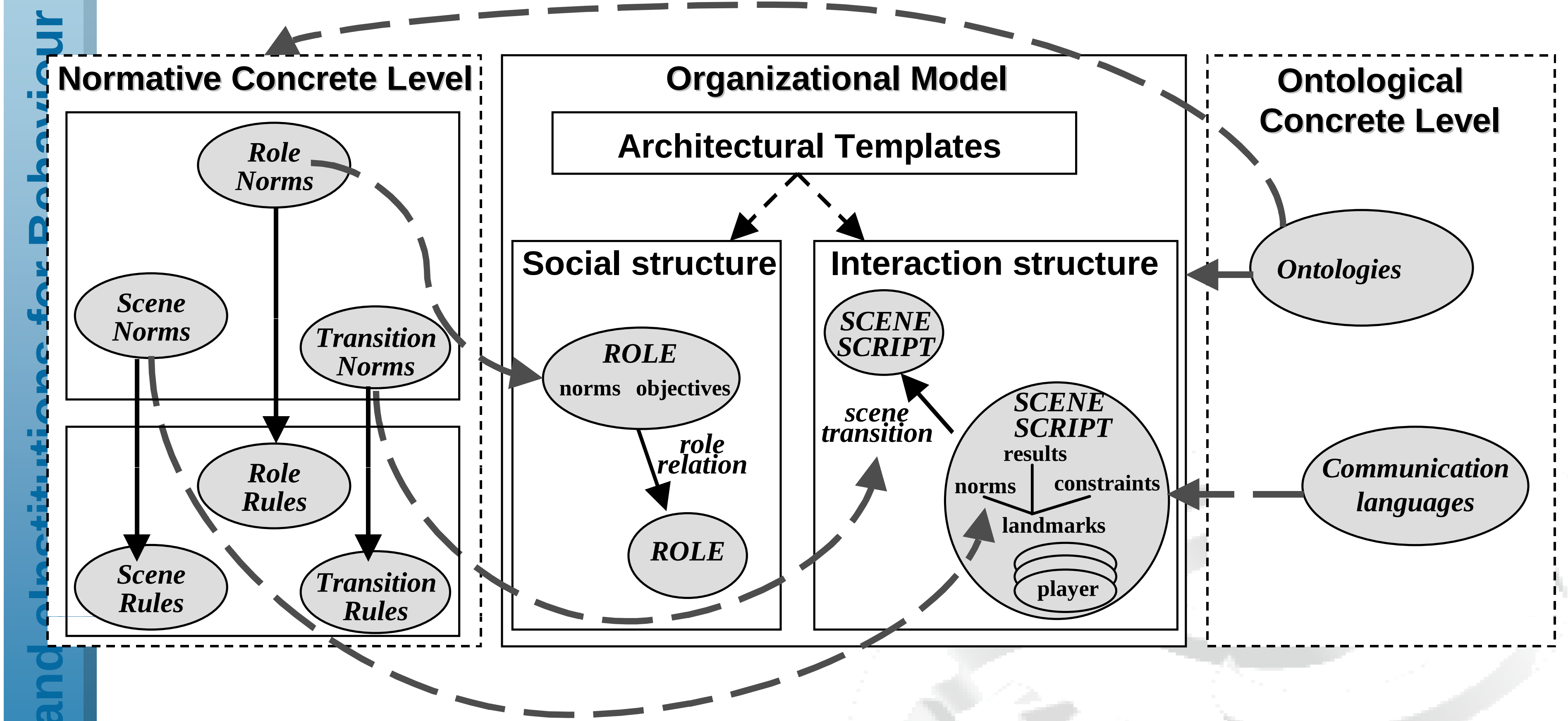
Electronic Institutions (III)

The **OMNI** framework

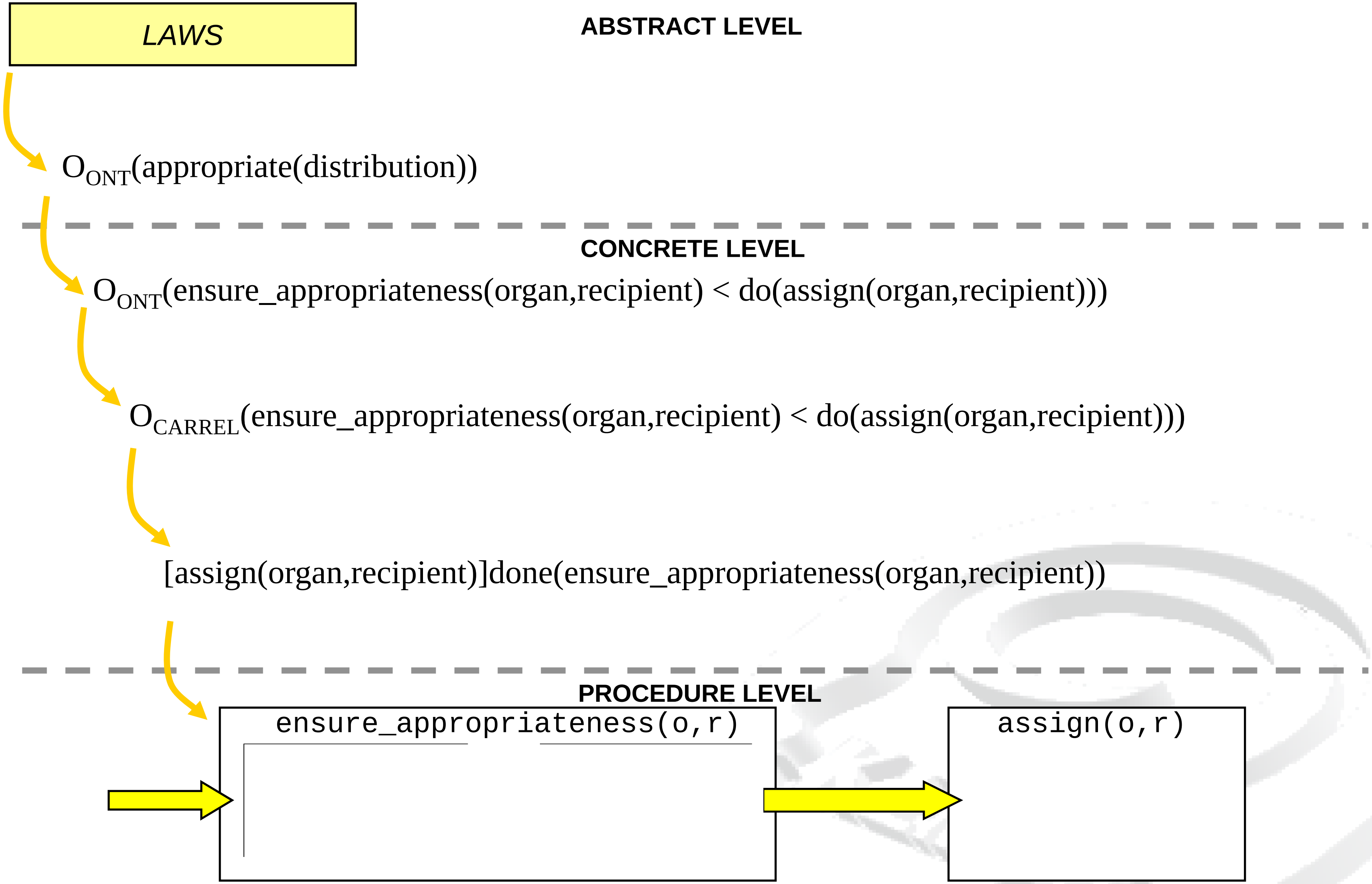


Electronic Institutions (III)

The **OMNI** framework



Example



LAWS

ABSTRACT LEVEL

O_{ONT}(appropriate(distribution))

CONCRETE LEVEL

O_{ONT}(ensure_appropriateness(organ,recipient) < do(assign(organ,recipient)))

O_{CARREL}(ensure_appropriateness(organ,recipient) < do(assign(organ,recipient)))

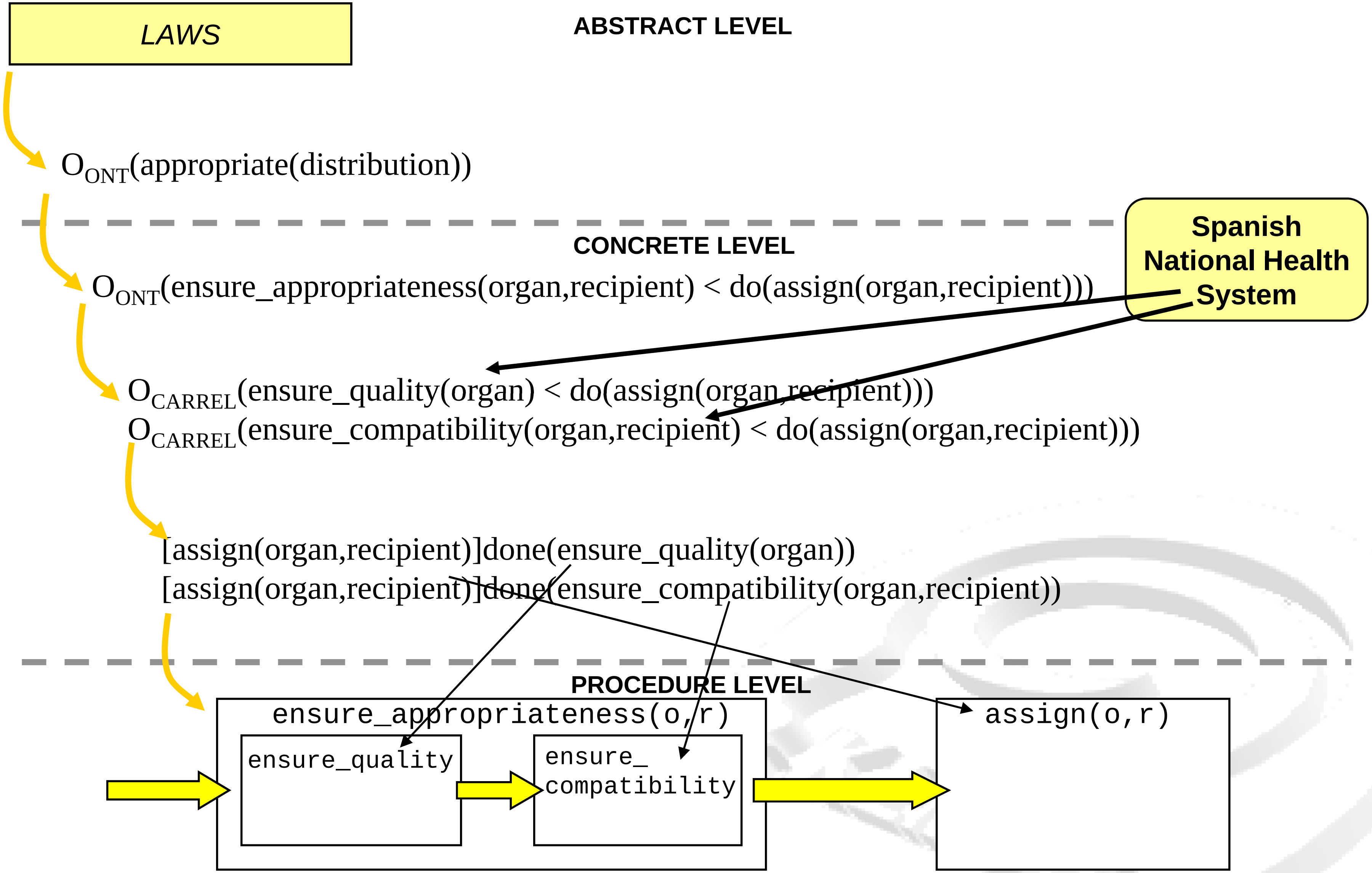
[assign(organ,recipient)]done(ensure_appropriateness(organ,recipient))

PROCEDURE LEVEL

```
ensure_appropriateness(o,r)
```

```
assign(o,r)
```

Context as source of interpretation



LAWS

ABSTRACT LEVEL

O_{ONT}(appropriate(distribution))

CONCRETE LEVEL

O_{ONT}(ensure_appropriateness(organ,recipient) < do(assign(organ,recipient)))

Spanish National Health System

O_{CARREL}(ensure_quality(organ) < do(assign(organ,recipient)))

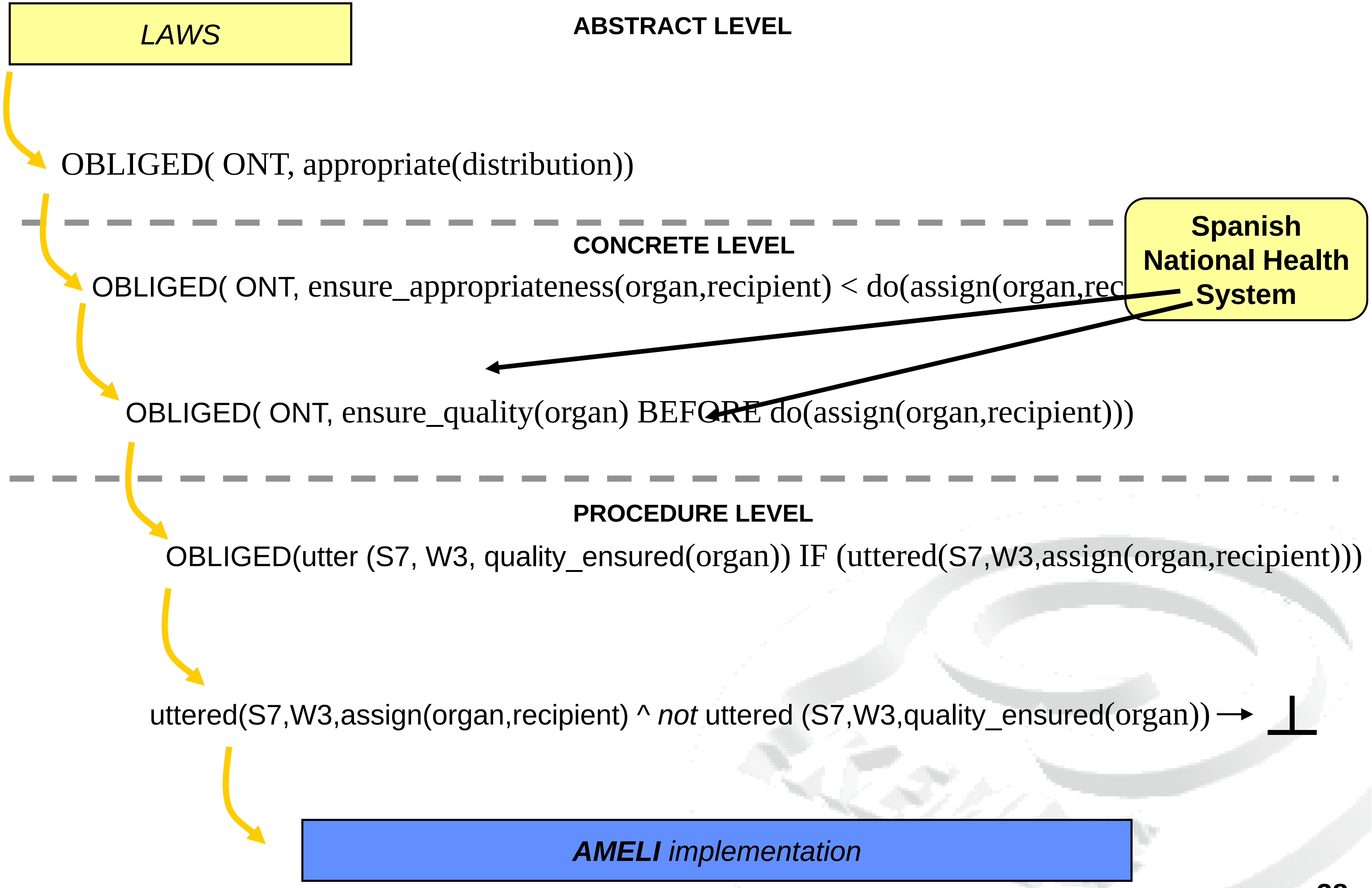
O_{CARREL}(ensure_compatibility(organ,recipient) < do(assign(organ,recipient)))

[assign(organ,recipient)]done(ensure_quality(organ))

[assign(organ,recipient)]done(ensure_compatibility(organ,recipient))

PROCEDURE LEVEL

Current version of the idea



LAWS

ABSTRACT LEVEL

OBLIGED(ONT, appropriate(distribution))

CONCRETE LEVEL

OBLIGED(ONT, ensure_appropriateness(organ,recipient) < do(assign(organ,rec

Spanish National Health System

OBLIGED(ONT, ensure_quality(organ) BEFORE do(assign(organ,recipient)))

PROCEDURE LEVEL

OBLIGED(utter (S7, W3, quality_ensured(organ)) IF (uttered(S7,W3,assign(organ,recipient)))

uttered(S7,W3,assign(organ,recipient) ^ *not* uttered (S7,W3,quality_ensured(organ)) → ⊥

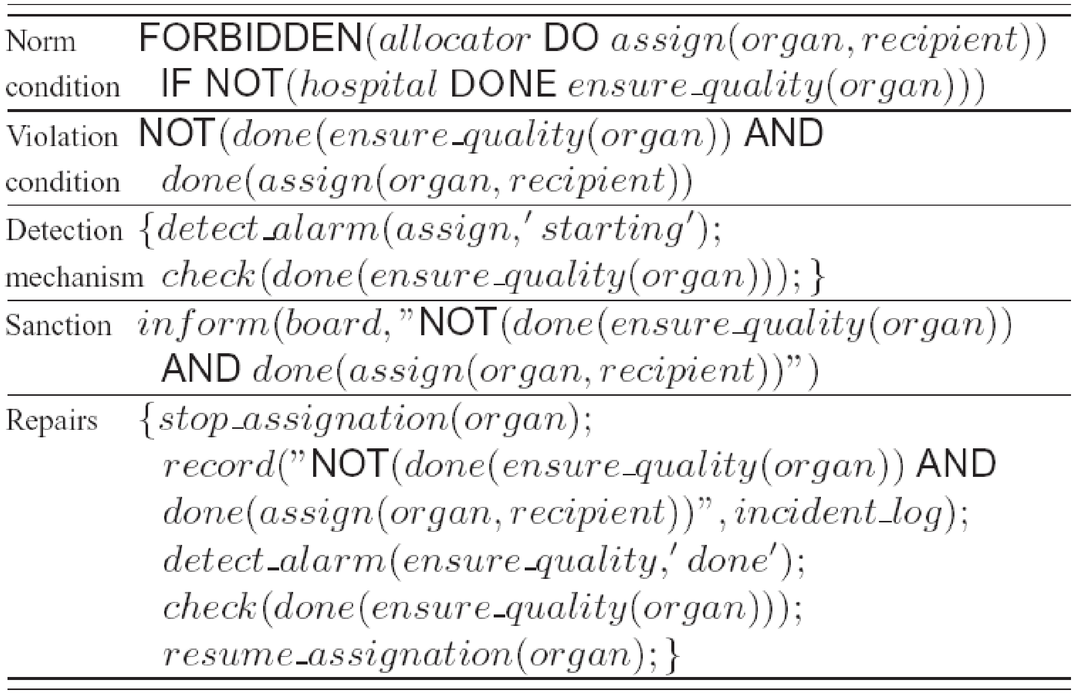
***AMELI** implementation*

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 controlable
    - 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
condition
	FORBIDDEN($allocator$ DO $assign(organ, recipient)$)
	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	$\{stop\_assignation(organ);$ $record("$NOT$(done(ensure_quality(organ))$ AND $done(assign(organ, recipient))", incident\_log);$ $detect\_alarm(ensure\_quality, 'done');$ $check(done(ensure\_quality(organ)));$ $resume\_assignation(organ);\}$

CONTRACT

SOA Governance
as
Contract-based Institutions

Knowledge Engineering and Machine Learning Group
UNIVERSITAT POLITÈCNICA DE CATALUNYA

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

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 → norms!)
  - This is, in fact, an electronic institution!

Contracting language overview (I)

Contract expressions

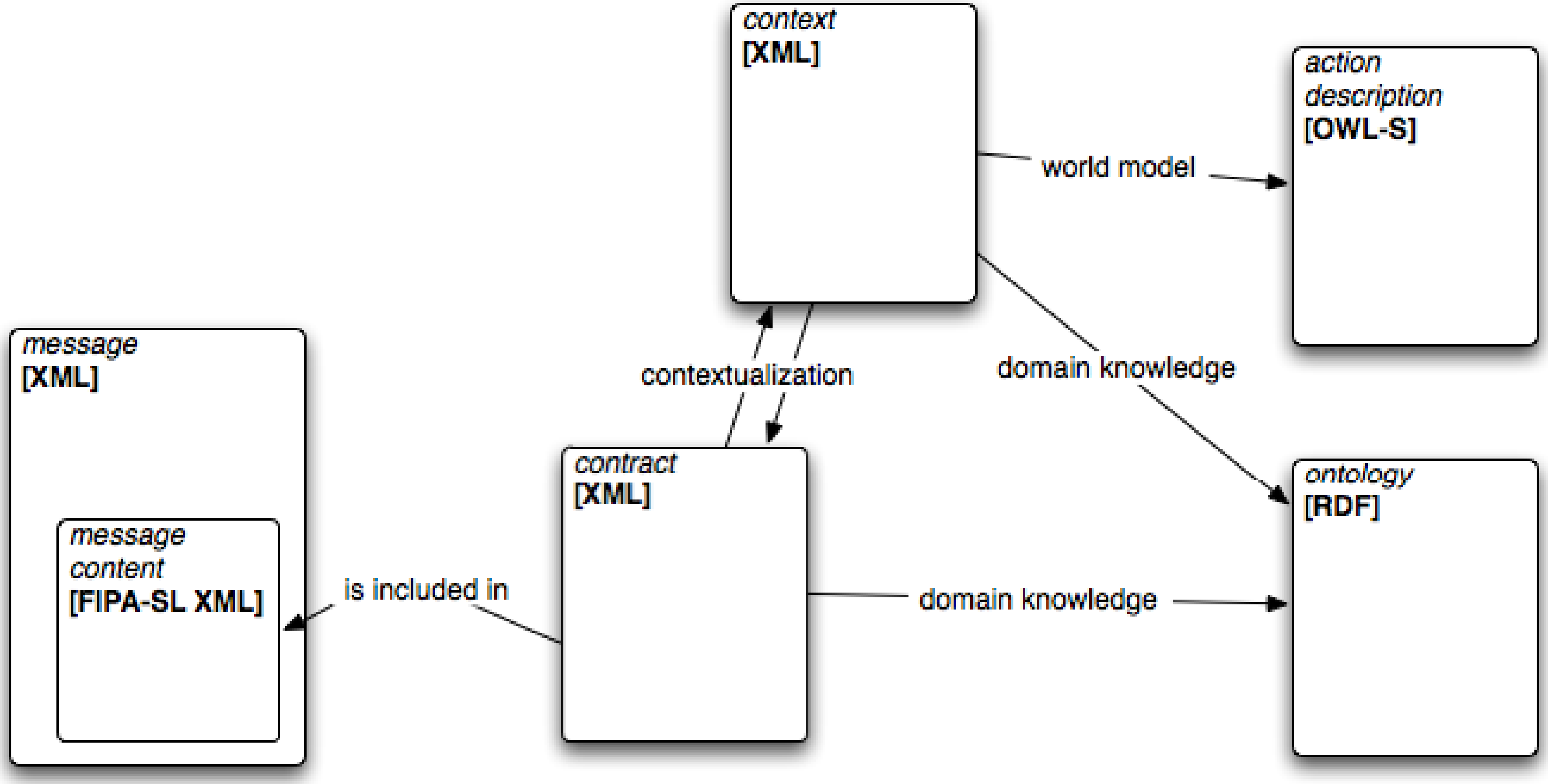
```
<ISTContract
            Contra
            Startir
            Ending
            xmlns:
            xsi:noN
  <Contextualiza
    ...
  </Contextualizat
  <Definitions>
    ...
  </Defi
  <Clauses
    ...
  </Clauses
</ISTContract>
```

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

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

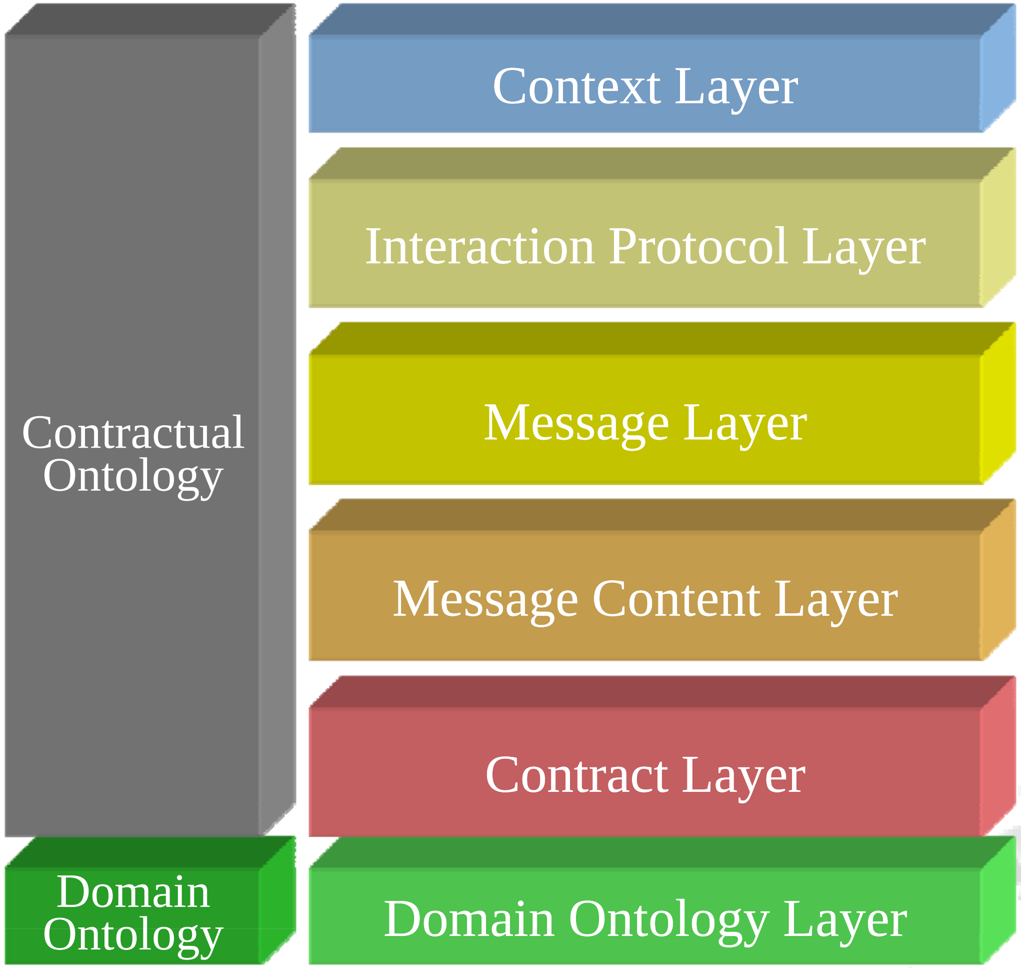
Contracting language overview (II)

Relations between language components



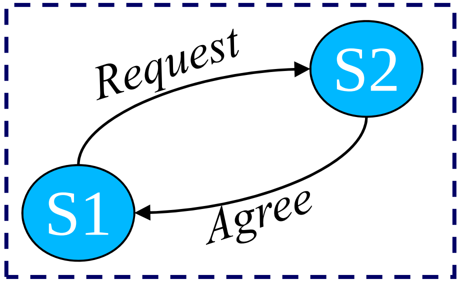
Contracting language overview (III)

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”

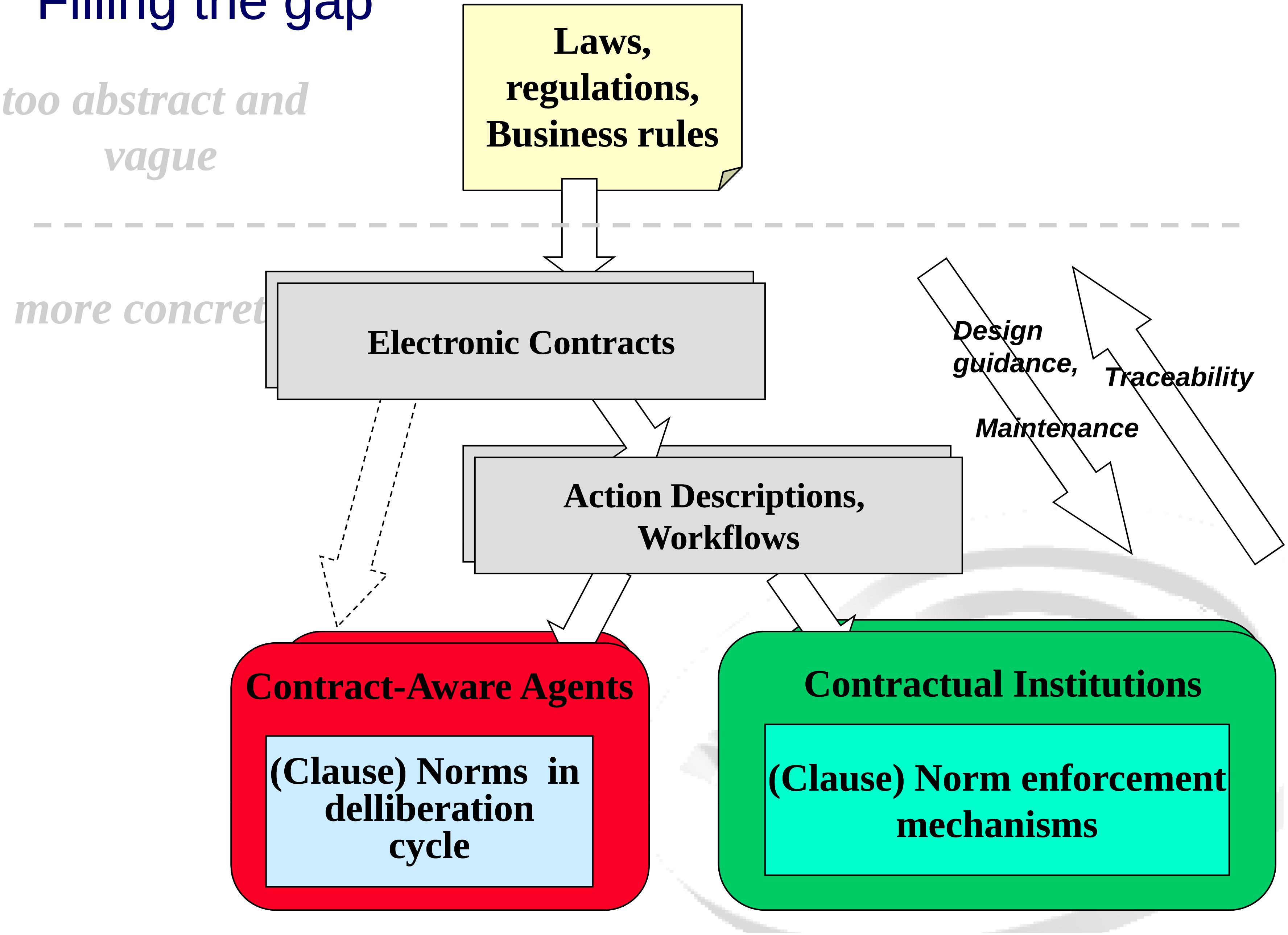
Domain terms: *car, workshop, repair*

Conclusions

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

Filling the gap



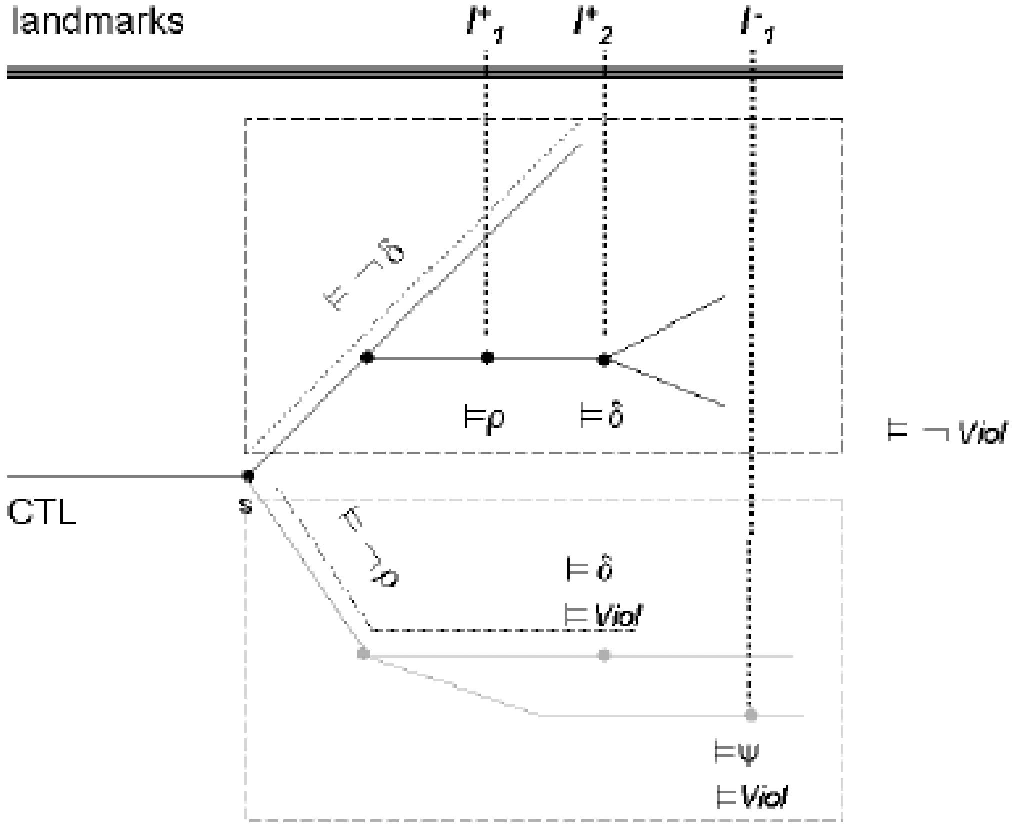
Ongoing work: using landmarks for formal connection

- 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.

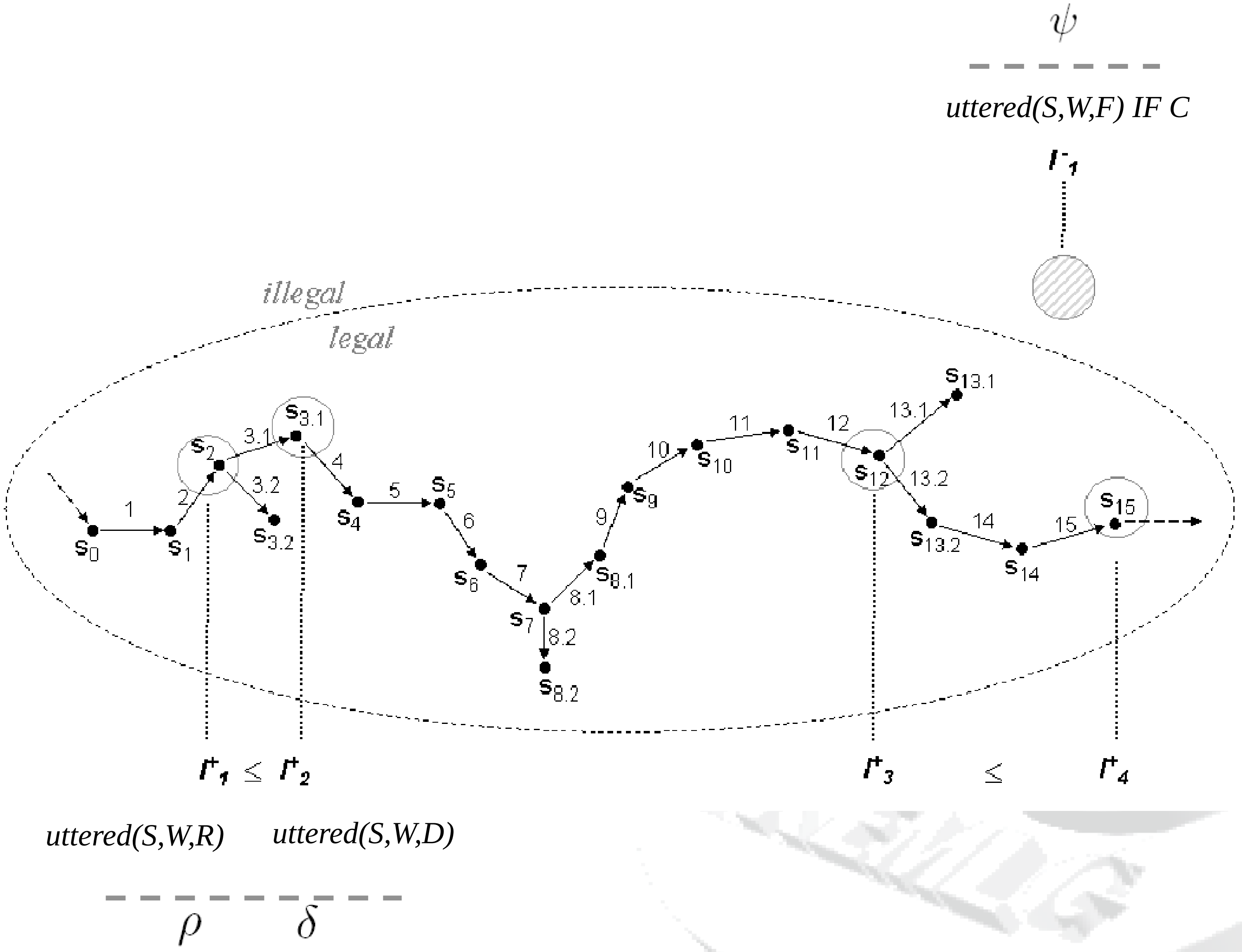
From Norms to Landmark Patterns

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

$$F\psi$$



From Landmark Patterns to Protocols



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