4. Multiagent Systems Design
Part 5:
Coordination models:
Social Models

Electronic Institutions

Javier Vázquez-Salceda
SMA-UPC

Introduction to Electronic Institutions

- Institutions as Social Structures
- e-Institutions
- Approaches in Literature
Introduction

- Open multi-agent systems (MAS) have to cope with several issues
  - Heterogeneity among members
  - Communication
  - Participants’ trust
  - Coordination
  - Cooperation

**Agent Autonomy VS Control**

- Idea: multi-agent design can benefit from social abstractions
  - Study the problem from the societal and the individual points of view.

Institutions as Social Structures

- **Social Structures** define a social level to enhance coordination by means of interaction patterns
- **Institutions** are a kind of social structure where a corpora of constraints shape the behaviour of the members of a group
- The definition of a (human) Institution uses to include
  - Regulations about the interactions
  - Conventions: ‘institutional facts’ vs ‘brute facts’
    - e.g. 1: ‘murder’ (vs killing)
    - e.g. 2: ‘incest’ (vs sexual act)
    - e.g. 3: ‘ownership’ (vs physical possession)
    - e.g. 4: ‘marriage’ (vs living together)
  - Procedures and protocols for creating and determining institutional facts
Introduction

e-Institutions

- An e-Institution is the computational model of an institution through
  - The specification of the institution's norms in (some) suitable formalism(s).
  - The specification of the institution’s procedures and protocols

- In the context of MAS they:
  - 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)

Agent behaviour guided by Norms

Introduction

Why a Language for Norms?

Laws, regulations

Language for norms (Formal & Computational)

Electronic Institutions

Norm enforcement mechanisms

Normative Agents

Norms in deliberation cycle
Norms from the Agent perspective

- Influence of norms in Agent behaviour
- Possible World semantics
- Norms in the agent interpreter

Norms and Agents (I)
Influence of norms in the BDI deliberation cycle

- Agent
- Sensors
- Perception

State

KB

Goals

How is the world now?

What if I perform action A?

Which action do I choose?

(input)

Environment

Actuators

Action

norms
(offerations, permissions...)

What if I perform
action A?

What is the effect of actions?

How the world evolves?

percpetion

state

goals
Norms and Agents (II)

- How do norms influence the behavior of the agent?
  - Agent has no knowledge about norms
  - Norms are built into the agent’s code
  - Norms are built into the plans and protocols the agent uses
  - Norms are explicit elements in the agent’s reasoning
    - Agent may or may not adopt the norms
    - Agent may or may not follow the norms
      - agent follows the norm whenever possible
      - agent violates the norm sometimes
      - agent violates the norm always if possible

Norms and Agents (III)

- Problems:
  - Norms are more abstract than the procedures
  - 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”
Norms and Agents (IV)

- 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.

Norms and Agents (V)
Possible Worlds Semantics

[Diagram showing possible worlds semantics with sets B₁, K₁, L₁, N₁, and N_w connected through arrows indicating relationships and roles.]
Norms and Agents (V)

Legally accessible worlds

- 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 may include the actions to recover the system from a violation.

Norms and Agents (VI)

- **Problem:** in this model interpretation of norms is completely done by the agents.
  - How to ensure that two agents that play the same role have similar sound interpretations of the norms that apply to them?
- **Solution:** to fix part of the interpretation in a given context.
- **Idea:** Agents do not have a relation with the WHOLE world but a part → **context** of an agent.
Norms and Agents (VII)

- **Context** is a set of worlds with a shared vocabulary and a normative framework.
- **Effects on the Normative Dimension**
  - The generic norms applied to the world as a whole are called **Abstract Norms**.
  - **Concrete Norms** are interpretations of the abstract norms in a given context.

\[
CN_a = I(AN_w, C_a) \\
CN_b = I(AN_w, C_b)
\]

Norms and Agents (VIII)

\[
R_a
\]

role

4. Multiagent Systems Design
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. $B := br(B, \rho)$;
6. $D := \text{options}(B, I)$;
7. $I := \text{filter}(B, D, I)$;
8. $\pi := \text{plan}(B, I)$;
9. while not (completed or succeeded(I, B)) do
10. $\alpha := \text{do}(\pi)$;
11. $\text{execute}(\alpha)$;
12. $\pi := \text{tail}(\pi)$;
13. get next percept $\rho$;
14. $B := br(B, \rho)$;
15. if reconsider(I, B) then
16. $D := \text{options}(B, I)$;
17. $I := \text{filter}(B, D, I)$;
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
24. 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 from the Institutional perspective

- SMART normative model
- AMELI/ISLANDER
- HARMONIA
- OMNI
SMART Normative Systems model
Fabiola Lopez y Lopez, Mike Luck and M. d'Inverno.

- It is based in the SMART agent specification framework presented by d'Inverno and Luck [8].
  - The framework defines concepts such as
    - objects,
    - agents (which are objects with goals)
    - autonomous agents (which are agents with motivations).
  - This framework is developed in the Z specification language [9].
- The SMART framework has been extended [2][7] to introduce, as part of the framework, representations of
  - norms
  - normative agent
  - Normative MAS
- The authors have also presented an architecture for autonomous social and normative agents in order to reason about norms.
SMART Normative Systems model

Norm definition

<table>
<thead>
<tr>
<th>Norm</th>
<th>normativegoals : P.Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>addressees : P.AgentName</td>
</tr>
<tr>
<td></td>
<td>beneficiaries : P.AgentName</td>
</tr>
<tr>
<td></td>
<td>context : EnviState</td>
</tr>
<tr>
<td></td>
<td>exceptions : EnviState</td>
</tr>
<tr>
<td></td>
<td>rewards : P.Goal</td>
</tr>
<tr>
<td></td>
<td>punishments : P.Goal</td>
</tr>
</tbody>
</table>

\[\text{normativegoals} \neq \emptyset \]
\[\text{addressees} \neq \emptyset \]
\[\text{context} \neq \emptyset \]
\[\text{context} \cap \text{exceptions} = \emptyset \]
\[\text{rewards} \cap \text{punishments} = \emptyset \]

SMART Normative Systems model

Norm addressees, legislators, defenders and promoters

- Norms are related not only with the agents that should fulfil it or enforce it, but also with agents such as the one that issued the norm, the one(s) responsible of its enforcement (the **defenders**, the one that modified it or the one(s) that may be affected by a non-compliance of the norm.

<table>
<thead>
<tr>
<th>AuthoritiesNMAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>NormativeNMAS</td>
</tr>
<tr>
<td>legislators : P.AgentName</td>
</tr>
<tr>
<td>defenders : P.AgentName</td>
</tr>
<tr>
<td>promoters : P.AgentName</td>
</tr>
<tr>
<td>( \forall lg : \text{legislators} \Rightarrow (\exists \text{norm} : \text{legislation} &amp; lg \in \text{norm.addressees}) )</td>
</tr>
<tr>
<td>( \forall df : \text{defenders} \Rightarrow (\exists \text{norm} : \text{enforcement} &amp; df \in \text{norm.addressees}) )</td>
</tr>
<tr>
<td>( \forall pm : \text{promoters} \Rightarrow (\exists \text{norm} : \text{reward} &amp; pm \in \text{norm.addressees}) )</td>
</tr>
</tbody>
</table>
SMART Normative Systems model

Norm Lifecycle

- Norms are not modelled as static constraints but as objects that can have several states (such as issued, active, modified, fulfilled or violated).

![Diagram of SMART Normative Systems model]

```
SMARTNormativeModel
allinstances : NNormInstance
activenorms : NNormInstance
fulfillednorms : NNormInstance
unfulfillednorms : NNormInstance
rewardednorms : NNormInstance
punishednorms : NNormInstance

forall in : allinstances, (\exists n : generalnorms \land in\ instance (in, n))
forall na : activenorms, (\exists n, environment : activenorm (na, environment))
forall fn : fulfillednorms, (\exists n, environment : fulfilled (fn, environment))
forall unfn : unfulfillednorms, (\exists n, environment : unfulfilled (unfn, environment))
forall rn : rewardednorms, (\exists n, environment, rewardnorms)
(\exists n, environment : rewardnorm (rn, environment)) \land fulfilled (rn, environment))
forall pn : punishednorms, (\exists n, environment, enforcenorms)
(\exists n, environment : enforcenorm (pn, environment)) \land fulfilled (pn, environment))
```
SMART Normative Systems model
Norms and the concept of Power

- They also have analysed in [7] the different power relations that may arise in an agent society,
  - *institutional power*: social structures define norms that entitle agents to direct the behaviour of others (*institutional power*)
  - *personal power*: the power of an agent given by its capabilities to satisfy goals and the power of other agents to benefit or to hinder those goals.
- However,
  - no implementation of the architecture applying it to a real problem has been reported in literature,
  - there are no tools to support the development of a normative multi-agent system following their framework.

AMELI
M. Esteve, J.A. Rodríguez-Aguilar and P. Noriega

- AMELI [3] is an institution middleware that is based in an electronic institution specification (ISLANDER).
- The ISLANDER framework [4] [3] is composed of:
  - A Dialogical Framework
    - Linguistic and social structure (*roles*) to give meaning to agent interactions,
  - A Performative Structure
    - *scenes* and relationships between scenes (navigation, precedence, causality)
  - Rules
    - Role-dependent conventions to establish social commitments
- Two hypotheses:
  - All agent actions are *messages*, observable by the e-institution
  - An Agent should *never* break the norms.
AMELI
ISLANDER: Performative Structure (I)

Scene conversational graphs: Reception Room

AMELI
ISLANDER: Performative structure (II)

- Scene conversational graphs: Reception Room
AMELI
Role of the AMELI middleware

- The AMELI middleware aims to:
  - Mediate and facilitate agent communication within conversations (scenes).
  - Coordinate and enforce:
    - to guarantee the correct evolution of each conversation (preventing errors made by the participating agents by filtering erroneous illocutions, thus protecting the institution).
    - To guarantee that agents’ movements between scenes comply with the specification.
    - To control which obligations participating agents acquire and fulfil.
  - Manage information to facilitate participating agents the information they need to participate in the institution.

AMELI
Social Layer

- The current implementation of the social layer is composed of four types of agents:
  - An institution manager that starts the institution, authorises agents to enter, and controls the creation of scenes.
  - Scene managers responsible for governing scenes.
  - Transition managers control agents’ movements between scenes.
  - Governors devoted to mediate the interaction of an agent with the rest of agents within the institution and to control agents’ obligations.
    - Facilitates to the agent some information about the state of the institution.
    - Coordinates with other agents of the social layer for the correct execution of the institution.
    - Keeps track whether the agent pending acquires new obligations and fulfils some of its pending obligations.
AMELI

4. Multiagent Systems Design

AMELI

Limitations

- AMELI comes with a toolkit to help create institutional specifications, create the associated governors and create an e-Institution instance from an ISLANDER specification.

- But:
  - Weak notion of norms (norms only as restrictions, not as preference shapers).
  - Norms are never to be broken → no autonomy
  - Norms are only defined at the action-interaction level → too low-level
  - The only actions that can be controlled are messages
  - Message interactions are defined step by step → no flexibility.
  - Norm handling is done by the Governors, not the agents → agents cannot reason about the norms
  - No mechanism designed for the agents to introduce the norms in their reasoning → designer tends to hard-code them.

GOVERNORS
Agents Layer

Private Public
AMELI Social Layer

Communication Layer

INSTITUTION MANAGER SCENE MANAGERS TRANSITION MANAGERS

Institution Specification (XML format)
*HARMONIA*
J. Vázquez-Salceda

- Approaches in literature were **too theoretical** (e.g. SMART) or **too practical** (e.g. AMELI)
  - There was a gap between the specification of abstract norms and the concrete implementation inside *e-Organizations*

- **HARMONIA** [1] is a multi-level framework that proposes a formal connection between the different abstraction levels of a **Normative System**
  - It distinguishes between **normative** and **practical** levels
  - Analysis of **Context** in **Normative Systems**
    - *context and norms, nested contexts, super-contexts’ influence*
  - Establishes **connection** between formal specification and agent implementation
    - **top-down**: design guidance
    - **bottom-up**: track the origins of a protocol/plan
  - **Norm enforcement** as detecting illegal worlds

---

**E-INSTITUTION**

**BACKGROUND KNOWLEDGE**

**CONTEXT**

<table>
<thead>
<tr>
<th>STATUTES</th>
<th>ABSTRACT LEVEL</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Values</em></td>
<td></td>
</tr>
<tr>
<td><em>Objectives</em></td>
<td></td>
</tr>
<tr>
<td><em>Context</em></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INST. templates</th>
<th>CONCRETE LEVEL</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Abstract Norms</em></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RULE LEVEL</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Rule sets</em></td>
</tr>
<tr>
<td><em>Generic Policies</em></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PROCEDURE LEVEL</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Standards</em></td>
</tr>
<tr>
<td><em>Technologies</em></td>
</tr>
<tr>
<td><em>Algorithms</em></td>
</tr>
</tbody>
</table>

Policy implementation
The statutes define the
- Values
- Objectives
- Context
of the organization.

**E.g.: Organización Nacional de Trasplantes:**

The main objective of ONT is to increase the number of organ donations and the subsequent increase in available organs for transplants. The ONT operates according to the regulation of the national health system and it strives to distribute the donated organs in the most appropriate and correct way according to the current technical knowledge and according to the ethical principles of equality.

**HARMONIA**

**Abstract Level**

- Values are beliefs that we have about what is important and thus about how things should be.
  - "Appropriate distribution"
  - "Distribution according to ethical principles of equality"
  - "Fairness of transaction"
  - "Respect privacy of persons"

- Values can be considered as the most abstract level on which norms are expressed.
- The values of an organization can be defined by the set of Abstract Norms specified within the org. that contributes to that value

D(equity) =:= \{F_{ONT}(\text{discriminate}(x,y,\text{age})), O_{ONT}(\text{find_best_recipient}(\text{organ})), \ldots\} \\
D(\text{appropriate}(\text{distribution})) =:= O_{ONT}(\text{appropriate}(\text{distribution}))
HARMONIA
Abstract Level: Abstract Norms (II)

- **Problem:** These norms are too abstract, as they use concepts that are not fully described in the organization's ontology.
  - "It is forbidden to discriminate based on age"
    \[ F_{\text{ONT}}(\text{discriminate}(x,y,\text{age})) \]

- 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

HARMONIA
Abstract Level: Abstract Norms (III)

- **Example 1:** Abstract actions
  
  "a living donor should consent to the donation of an organ"

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

- **Example 2:** Vague terms
  
  "the ONT is obliged to ensure that the distribution of organs and tissues is appropriate"

  \[
  O_{\text{ONT}}(\text{ensure_quality}(\text{organ})) \land O_{\text{ONT}}(\text{ensure_compatibility}(\text{organ, recipient})) \implies F_{\text{ONT}}(\text{appropriate}(\text{distribution}))
  \]
**HARMONIA**

**Abstract/Concrete Level: Representing Norms**

- Formal representation of norms needed
- Which logic?
  - Abstract and Concrete Norms permit, oblige or prohibit
  - Concrete Norms may be conditional
  - Concrete Norms may have temporal aspects
  - Concrete Norms are relativized to roles

**variant of Deontic Logic**

---

**HARMONIA**

**Concrete Level: Concrete Norms**

- Concrete norms are the result of translating the abstract norms in the context of the organization into norms that make use of terms and actions that are defined in the organization’s ontology.

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

- **Problem:** HOW is a concrete norm like this implemented in an e-Organization?
**Rule Level (I)**

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

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

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

---

**Rule Level (II)**

- **Rules, Violations and Sanctions**
  - Violation rules define violations
  - A **violation** is composed by
    - **pre-conditions**
    - **sanction**
    - **side effects**
  - **Pre-conditions** are used by **Police Agents** to detect violations.
  - **Sanctions** are used by **Flexible Normative Agents** to reason about the utility of breaking the related rule.
  - **Side effects** are used by **internal agents** to recover the system from the violation.
**HARMONIA**

**Procedure Level**

- Idea: the final implementation of the system

- Formally, translation from Dynamic Logic to a **Procedural Language**:

  \[ \{P\} a \{Q\} \equiv P \rightarrow [a]Q \]

- Example:

  \[
  \begin{align*}
  \text{(not(done(assign(o, r))) \land done(ensure_appropriateness(o, r)))} \\
  \rightarrow \{\text{assign(o, r)}\} \text{ done(assign(o, r))}
  \end{align*}
  \]

  \[
  \downarrow
  \]

  \[
  \begin{align*}
  \{\text{not(done(assign(o, r)))} \& \text{ done(ensure_appropriateness(o, r))}\} \\
  \text{assign(o, r)} \\
  \{\text{done(assign(o, r))}\}
  \end{align*}
  \]

---

**HARMONIA**

**Connecting with the Procedure Level**

- **ABSTRACT LEVEL**

- **CONCRETE LEVEL**

- **RULE LEVEL**

- **PROCEDURE LEVEL**

---

4. Multiagent Systems Design
HARMONIA
Procedure Level: Implementing Norms in eInstitutions (I)

- Implementation of norms from institutional perspective
- 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

HARMONIA
Procedure Level: Implementing Norms in eInstitutions (II)

- Norm enforcement is not centralized but distributed in a set of agents, the Police 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 Police Agents, providing fast, very efficient aids for norm enforcement as additional platform services and mechanisms.
- A) Detection of the occurrence of an action
  - Police Agents may become overloaded checking ALL actions
  - **black list mechanism** (of actions to monitor) e.g., assign
  - **action alarm mechanism** (alarm to the Police Agent)
  - The Police Agent checks if conditions for a violation apply.
**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 Police Agents’ computation burden.

**C) Deadline control**

- A **clock trigger mechanism** to detect that a deadline has passed.

---

**Example of Norm violation definition**

<table>
<thead>
<tr>
<th>Norm</th>
<th>FORBIDDEN(allocation DO assign(organ, recipient))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condition</td>
<td>IF NOT(hospital DONE ensure_quality(organ)))</td>
</tr>
<tr>
<td>Violation</td>
<td>NOT(done(ensure_quality(organ))) AND</td>
</tr>
<tr>
<td>Condition</td>
<td>done(assign(organ, recipient))</td>
</tr>
<tr>
<td>Detection</td>
<td>{detect_alarm(assign, 'starting');</td>
</tr>
<tr>
<td>mechanism</td>
<td>check(done(ensure_quality(organ))))};</td>
</tr>
<tr>
<td>Sanction</td>
<td>inform(board, &quot;NOT(done(ensure_quality(organ))</td>
</tr>
<tr>
<td></td>
<td>AND done(assign(organ, recipient)))&quot;)</td>
</tr>
<tr>
<td>Repairs</td>
<td>{stop_assignation(organ); record(&quot;NOT(done(ensure_quality(organ)) AND</td>
</tr>
<tr>
<td></td>
<td>done(assign(organ, recipient))&quot;), incident.log);</td>
</tr>
<tr>
<td></td>
<td>detect_alarm(ensure_quality, 'done');</td>
</tr>
<tr>
<td></td>
<td>check(done(ensure_quality(organ)));</td>
</tr>
<tr>
<td></td>
<td>resume_assignation(organ);}</td>
</tr>
</tbody>
</table>
**HARMONIA**

All levels: Roles (I)

- Role definition guided by goal distribution
  - Origin: Objectives in Statutes
- Distribution of responsibilities

---

**Diagram:**

- Seek donations
- Increase donations
- Give consent
- Identify possible donors
- Manage tissue requests and offers
- Distribute tissues
- Request tissues and coordinate tissue transplantation
- Monitor the process
- Offer tissues

---

**Diagram:**

- Seek donations
- Increase donations
- Give consent
- Identify possible donors
- Manage tissue requests and offers
- Distribute tissues
- Request tissues and coordinate tissue transplantation
- Monitor the process
- Offer tissues

---
HARMONIA
All levels: Roles (I)

- Role definition guided by goal distribution
  - Origin: Objectives in Statutes
- Distribution of responsibilities

4. Multiagent Systems Design
**HARMONIA**

All levels: Roles (II)

- Role hierarchy extended with *power relations* to model the distribution of responsibilities not defined in the hierarchy.

---

**HARMONIA**

All levels: Context (I)

- **Statutes** make reference to a surrounding context.
- Links with the idea of *nested contexts*:
  - $e$-org$_X$ is a context defining a vocabulary and a normative system.
  - There are super-contexts that have an influence in $e$-org$_X$ definition.
- Formal view: influence as interpretation in the subcontext:
  - Counts-as operator $\Rightarrow$ as a link between interpretations.
- Influence in several levels of abstraction:
  - Vocabulary (terms, predicates).
  - Values, norms, rules and procedures.
4. Multiagent Systems Design

**HARMONIA**

All levels: Context (II)

D(appropriate(distribution))

**ABSTRACT LEVEL**

O\textsubscript{ONT}(appropriate(distribution))

**CONCRETE LEVEL**

O\textsubscript{ONT}(ensure_appropriateness(organ,recipient) < do(assign(organ,recipient)))

O\textsubscript{CARREL}(ensure_appropriateness(organ,recipient) < do(assign(organ,recipient)))

**RULE LEVEL**

(assign(organ,recipient))\textsubscript{done}(ensure_appropriateness(organ,recipient))

**PROCEDURE LEVEL**

ensure_appropriateness(\textit{o},\textit{r}) → assign(\textit{o},\textit{r})

57
HARMONIA
All levels: Background Knowledge

- The Background Knowledge is a repository containing templates that can be adapted to create new e-organizations
  - At the abstract level, it provides a collection of abstract norms and the related ontology and abstract roles
  - At the concrete and rule levels, it provides templates for some generic policies
    - e.g., the security policy
    - concrete norms, rules and ontology
  - At the procedure level it provides a link with the standards, technologies and algorithms needed to implement the policies

- Idea for future: Institutional templates to be parameterized, adapted or implemented to build e-Institutions.

OMNI
J. Vázquez-Salceda, V. Dignum and F. Dignum

- OMNI: Organizational Model for Normative Institutions [5]
  - Integration of HARMONIA’s Normative concepts [1] with OperA’s Organizational concepts [6]
  - integrated framework for both
    - closed systems with fixed participants and interaction protocols,
    - open, flexible systems that allow and adapt to the participation of heterogeneous agents with different agendas.

- Layered Approach
  - Abstract Level: Requirement analysis
  - Concrete Level: Analysis and design process
  - Implementation Level: Design specification in a given multi-agent architecture

- Top-down
  - Methodological guidance for design

- Bottom-up
  - Trace of origin and motivation for rules and protocols
OMNI
Levels and dimensions

OMNI
Dimensions

- **Normative Dimension**
  - Norms and rules that guide agent behaviour
  - Includes a model of the environment regulations
  - Comes from HARMONI/A

- **Organizational Dimension**
  - Captures the organizational structure and requirements
  - Comes from OperA, with norm language coming from HARMONI/A

- **Ontological Dimension**
  - Content: concepts and relationships
  - Communication Language
  - Comes from OperA
OMNI
Abstract Level

- **Statutes**
  - Main objective(s) of the organization,
    - Guides organizational design
    - Input for the Organizational Model
  - Values that direct the fulfilling of this objective
    - Guides normative design
    - Input for the Norm level
  - Context
    - Guides the ontological design
    - Influences also the normative design

- **Generic Terms**
  - In-contextual concepts

- **Model Ontology**
  - concepts of the framework itself
  - E.g. norm, rule, role, group, violation, landmark...

OMNI
Concrete Level

- **Analysis and design process**
  - Based on abstract values and objectives

- **Refinement in three dimensions**
  - **Organizational Model**
    - Social Structure
      - roles, groups, dependencies
    - Interaction Structure
      - scene scripts, connections, transitions
  - **Normative Structure**
    - role, scene and transition norms (deontic)
    - role, scene and transition rules (operational)
  - **Communication Structure**
    - Concrete ontological concepts, communicative acts, domain ontology
OMNI
Concrete Level

Normative Concrete Level:
- Role Models
- Scene Norms
- Transition Rules

Organizational Model:
- Architectural Templates
- Social Structure
- Interaction Structure

Ontological Concrete Level:
- Ontologies
- Communication Languages

4. Multiagent Systems Design

OMNI
Concrete Level: Social Structure
- Role Model based in goal decomposition coming from objectives in statutes

Conference organization
- Conference society
- Conference management
- Constituency roles
- Internal roles
- External roles
- Coordinator role
- Session chair role
- Author role
- PC member role
- Presenter role

Coordination of all activities
- Session coordination
- Write paper
- Review paper
- Present paper
OMNI
Concrete Level: Social Structure. Role dependencies

Role: PC Member

<table>
<thead>
<tr>
<th>Objectives</th>
<th>paper_reviewed(P, Rep)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sub-objectives</td>
<td>{ read(P), report_written(P, Rep), review_received(Org, P, Rep) }</td>
</tr>
<tr>
<td>Rights</td>
<td>access-confirman-program(me)</td>
</tr>
<tr>
<td>Norms</td>
<td>OBLIGED understand(English)</td>
</tr>
<tr>
<td></td>
<td>IF DONE assigned(P, me, Deadline) THEN OBLIGED paper_reviewed(P, Rep) BEFORE Deadline</td>
</tr>
<tr>
<td></td>
<td>IF DONE paper_assigned(P,me,_) AND direct_colleague(author(P)) THEN OBLIGED review_refused(P) BEFORE TOMORROW</td>
</tr>
<tr>
<td>Type</td>
<td>external</td>
</tr>
</tbody>
</table>

**OperA** Role descriptions + **HARMONIA** norm language
OMNI
Concrete level

OMNI
Concrete Level: Interaction structure
OMNI
Concrete Level: Interaction Structure. Scene example

<table>
<thead>
<tr>
<th>Interaction Scene: Review Process</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Roles</strong></td>
</tr>
<tr>
<td><strong>Results</strong></td>
</tr>
<tr>
<td><strong>Interaction Patterns</strong></td>
</tr>
<tr>
<td><strong>Norms</strong></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

**OMNI**
Implementation Level

- **Norm enforcement**
  - **Protocols** and **Rules**: enable agents to comply with organizational norms
- **Role enactment**
  - **Social Contracts**: commitments regulating the enactment of roles by individual agents.
  - **Interaction Contracts**: specific interactions such as agreed upon by agents
- **Ontologies**
  - **Specific Communication Acts**: actual communication languages actually as fixed in interaction contracts.
  - **Specific Communicative Acts**: implement the content ontologies.
OMNI
Implementation Level: Social Model

- Role enacting agents (rea)
  - agents realizing expected behavior of role
- Social Contract
  - describes a specific agreement for a rea

social-contract(agent1, PC-member, {})
\( \forall \text{scene: PC-member } \in \text{roles(scene), rea(agent1, PC-member, scene)} \)

social-contract(agent2, PC-member, {} maximum to review papers is 3)\)
\( \forall \text{scene: PC-member } \in \text{roles(scene), rea(agent2, PC-member, scene)} \)

OMNI
Implementation Level: Interaction Model

- Scene Instantiation
  - protocols realizing landmarks
- Interaction contract
  - describes a specific performance of a scene

interaction-contract( {PC-Chair, pc1, pc2, pc3, pc4}, review-process, {}, P1)

interaction-contract( { PC-Chair, pc3, pc5}, review-process,
  { IF NOT reviews-done BEFORE DeadlineR THEN PERMITTED(PC-Chair, paper-accepted(P)}, P2)
OMNI
Implementation Level: Norm Enforcement

- Based in HARMONIA
- New idea: OperA’s Landmarks to monitor norm execution
  - Landmark as observable system state

References


References

