Uma lógica para a especificação formal de organizações

Olga Pacheco
CCTC/Departmento de Informática
Universidade do Minho, Portugal

Seminários do MICEI
Novembro de 2005

Questão

Discuta em que medida a especificação normativa de uma organização pode ser útil a essa organização e às pessoas que com ela interagem.

Contexto

Uma lógica para a especificação formal de organizações

- Lógica modal
  - $\Box\psi$ - é necessário que ψ seja verdade.
  - $\Diamond\psi$ - é possível que ψ seja verdade.
  - Mundos possíveis: estados de informação ao longo do tempo, do espaço, de níveis de conhecimento, ...

- Modos de ser verdade:
  - Lógica temporal: $\Box\psi$ - ψ verifica-se sempre no futuro, $\Diamond\psi$ - ψ verifica-se eventualmente no futuro.
  - Lógica deôntica: $O\psi$ - é obrigatório que ψ se verifique, $P\psi$ - é permitido que ψ se verifique.
  - Lógica epistémica: $K_a\psi$ o agente a sabe que ψ se verifica.
  - ...

Contexto

Uma lógica para a especificação formal de organizações

- Especificação: descrição da estrutura e comportamento.
- Em sistemas de software complexos podemos não ter o controlo total sobre o comportamento do seus componentes:
  - informação incompleta
  - demasiado complexa
  - custos incomportáveis
  - intervenção humana imprevisivel
  - ...
- Mas, pelo menos, deve saber-se qual o comportamento ideal/esperado de cada componente.

Contexto

- Especificação normativa:
  - normas descrevendo o comportamento esperado do sistema e dos seus componentes.
  - assumindo que podem ocorrer falhas (violações das normas definidas - comportamento efectivo diferente do comportamento esperado)
  - e definindo como reagir a falhas (sanções, recuperação do estado ideal).

- Normas: conjunto de obligações e permissões.

- Normas regulam acções de agentes - o seu comportamento.

Temos de relacionar obrigações e permissões com as acções dos agentes.

Um exemplo

Caracterização da noção de confiança em sistemas computacionais, inseridos num contexto organizacional .

(Tendo por base o seguinte artigo:
Olga Pacheco, *Normative specification: a tool for trust and security*, Proceedings of the $3.^{rd}$ International Workshop on Formal Aspects of Security and Trust (FAST'05), Newcastle, Julho de 2005)

Introduction: *trusty computer system*

- People trust computer systems
  - if they don't fail or
  - if they do, by believing that someone will be responsible by any damage caused.
- But we cannot attribute responsibilities to a software entity!
- There must always be some person responsible (human person or artificial person).

How to establish this link between software entities and persons?

Introduction

We propose an unifying and integrating model of organizational systems, where software agents are specified at the same level as human agents.

As agents may exhibit non ideal behavior we will use normative specification to describe expected behavior.

We have to relate obligations and permissions with actions of agents, confronting expected with actual behavior.

We want to establish a responsibility link between software entities and persons, which will allow us to attribute responsibility for every action.

A model: *Artificial persons*

A company is

- an abstract entity
  (*We don't see a company walking in the street...*)
- a collective entity
  (*A company "lives" through the persons that constitute it.*)

A model: *artificial persons*

A company is an abstract entity

- it must be classified in legal terms as an artificial person: association, foundation, liability society,...;

- that legal classification determines in global terms

  the structure of the company (a set of positions that the members of the company will occupy)
  a set of norms describing how the holders of each position should behave (what they are obliged to do, or not to do, what are their permissions, ...)

- The statutes of a company contain all the information about the structure and the norms that characterize it. They are public and describe the company's aims and how they will be achieved.

A model: *artificial persons*

Any artificial person has

- juridical personality:
  it may be the subject of obligations, permissions, rights...
- legal qualification:
  it can exercise its rights and be responsible for the unfulfillment of obligations.

A company is a collective entity:

- A company must act to fulfill its obligations.
- A company acts through the agents that "support" its structure.
- It must be defined:
  - ∗ How the obligations of the company are transmitted to the agents that support its structure;
  - ∗ How the actions of the agents may "count as" actions of the company.

A model: *artificial persons*

Law imposes the legal classification of a company as an artificial person, for security reasons: people that interact with the company must know what to expect from the company and who is going to be responsible when things go wrong.

People trust (some) companies and interact with them, because they feel protected by law and know what they should expect of them.

From *artificial persons* to *institutional agents*

Based on the concept of artificial person we proposed a model for organizational systems: **institutional agent**.

Institutional agents are suited to model collective entities that have a role-based, stable structure which is supported by agents.

Agents always act in roles.

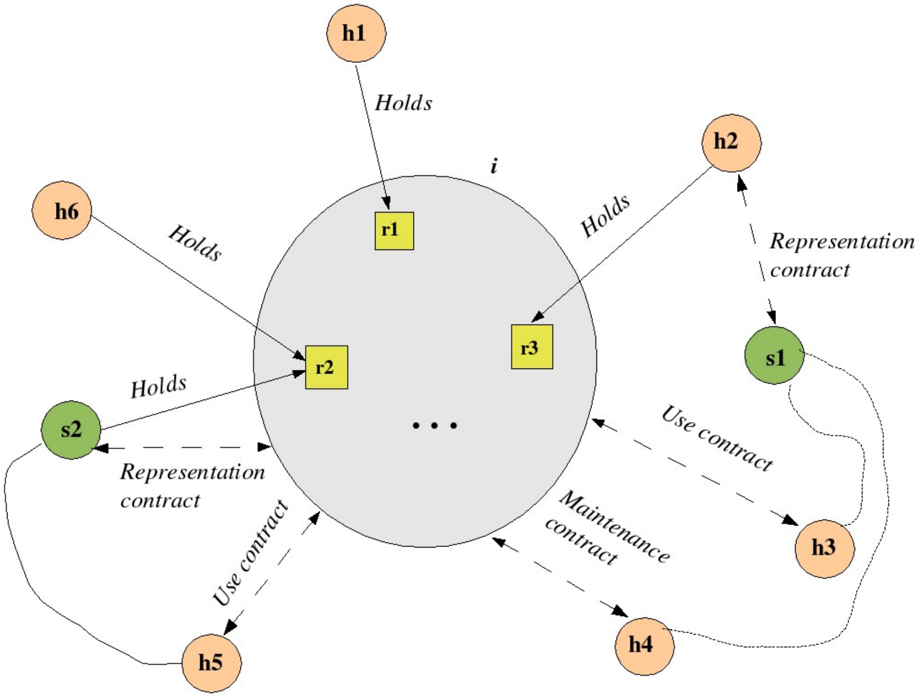
Why do we need roles?

- Roles provide the deontic context to evaluate actions: is an action permitted/forbidden/obligatory?
- The effects of an action depend on the role: the same action done by the same agent in different roles has different consequences.

Roles and action in a role

- Roles correspond to positions in an organization (president of the board, member of a program committee) or in a contract (representative, manager,...).
- Roles are occupied by agents (human, software or institutional agents)
- A role may be occupied by several agents.
- An agent may hold different roles.
- When an agent holds a role he inherits the deontic characterization of the role.

Institutional agents



Legend:

Agents: i, $h_1..h_6$, $s1$ and $s2$;

Roles: $r1$, $r2$ and $r3$

$$ST_i = < R_i, DCR_i, TO_i, RER_i >$$
$$SA = < iAg, sAg, hAg, CONT >$$

The logic $\mathcal{L}_{\mathcal{DA}}$

- We defined a first-order, multi-modal, many-sorted logic.
- Action operator: $E_{a:r}\ \phi$
  (agent a acting in role r brings about ϕ)
- Deontic operators:
  - $O_{a:r}\ \phi$ (agent a, when playing role r, is under the obligation ϕ)
  - $P_{a:r}\ \phi$ (agent a, when playing role r, is permitted to do ϕ)

The logic $\mathcal{L}_{\mathcal{DA}}$: *the formal language*

- Sorts:
  - Ag - agent sort
    iAg - institutional agent sort;
    sAg - software agent sort;
    hAg - human agent sort).
  - R - role sort $\quad (itself)$.
  - AgR - agent in a role sort.

- Some predicates:
  - $is-iAg$ of sort (iAg);
    $is-sAg$ of sort (sAg);
    $is-hAg$ of sort (hAg).
  - $is-rg$ of sort (Ag) $\quad (qual(a:r)$ abbreviates $is-r(a))$.

The logic $\mathcal{L}_{\mathcal{DA}}$: *some properties*

	Axioms:	
(T_E)	$E_{a:r}B \longrightarrow B$	success operator
(C_E)	$E_{a:r}A \wedge E_{a:r}B \longrightarrow E_{a:r}(A \wedge B)$	
(Qual)	$E_{a:r}B \longrightarrow qual(a : r)$	agents that act in roles are qualified
(Itself)	$(\forall_x) qual(x : itself)$	every agent is qualified to act as itself
	Proof rule:	
(RE_E)	If $\vdash A \leftrightarrow B$ then $\vdash E_{a:r}A \leftrightarrow E_{a:r}B$	

Axioms:	
(C_O)	$O_{a:r}A \wedge O_{a:r}B \longrightarrow O_{a:r}(A \wedge B)$
$(O \longrightarrow P)$	$O_{a:r}B \longrightarrow P_{a:r}B$
$(O \longrightarrow \neg P \neg)$	$O_{a:r}B \longrightarrow \neg P_{a:r}\neg B$
$(O \wedge P)$	$O_{a:r}A \wedge P_{a:r}B \longrightarrow P_{a:r}(A \wedge B)$
Proof rules:	
(RE_O)	If $\vdash A \leftrightarrow B$ then $\vdash O_{a:r}A \leftrightarrow O_{a:r}B$
(RM_P)	if $\vdash A \longrightarrow B$ then $\vdash P_{a:r}A \longrightarrow P_{a:r}B$
(RM_{EP})	If $\vdash E_{a_1:r_1}A \longrightarrow E_{a_2:r_2}B$ then $\vdash P_{a_1:r_1}A \longrightarrow P_{a_2:r_2}B$

Deontic characterization of roles

The deontic characterization of a role in an organization is part of the identity of the organization and does not depend on the agent that holds that role in a particular moment.

$$O_r\psi \overset{abv}{=} (\forall_x)(qual(x:r) \rightarrow O_{x:r}\psi)$$
$$P_r\psi \overset{abv}{=} (\forall_x)(qual(x:r) \rightarrow P_{x:r}\psi)$$

- When we have multiple agents holding a role, all of them "inherit" the deontic characterization of the role. For instance, if there is some obligation associated to a role, all of its holders will be under that obligation and all of them will have to fulfill it.

Representative roles

- To express the representation notion associated to an agent in a role, we introduce the abbreviation:

  $$(x : r_1) : REP(y : r2, \psi) \overset{abv}{=} (E_{x:r1}\psi \rightarrow E_{y:r2}\psi)$$

  Agent $x : r_1$ is representative of $y : r2$ for ψ means that when x acting on role $r1$ brings about ψ, this counts as y having produced ψ (in role $r2$).
  ψ is the scope of representation.

- To express the notion of representative role we use the abbreviation:

  $$r1 : REP(a : r2, \psi) \overset{abv}{=} (\forall_x)(E_{x:r1}\psi \rightarrow E_{a:r2}\psi)$$

  Any agent that holds role $r1$ and brings it about that ψ when acting in that role, produces ψ on behalf of a (acting in role $r2$).

Transmission of obligations

To express the transmission of obligations of an organization to specific roles of its structure (and indirectly, to the holders of those roles), we can use formulas like the following ones:

$O_{x:itself}\psi \rightarrow O_r\psi$ for r a role of the structure of organization x .

Contracts

Arbitrary contract:

$$\begin{array}{l} C(a,b) \;=\; \mathit{qual(a{:}r1)} \;\wedge\; \mathit{qual(b{:}r2)} \\ P_{a:r1}B \;\wedge\; P_{a:r1}C \;\wedge O_{b:r2}D \;\wedge \\ (a:r1):REP(b:r2,B) \;\wedge\; (a:r1):REP(b:r2,C) \end{array}$$

Titularity contract:

$$\begin{array}{l} C(a,i) \;=\; \mathit{qual(a{:}r)} \;\wedge \\ O_{a:r}B \;\wedge\; P_{a:r}C \;\wedge\; O_{i:itself}D \\ E_{a:r}\neg B \rightarrow O_{a:r}F \end{array}$$

Trusty institutional agents

- **How can we trace responsibilities?**
- Using contracts to explicitly state the relationships that exist between the agents:
  - A software agent must always act as representative of some other agent (the institutional agent or some other agent member of the institutional agent). We must state that in the “contract” between the organization and the software agent (that attributes the role it plays in the organization).
  - There must exist (formal or informal) contracts between the persons involved:
    - a contract between the company and the software developer (to assure maintenance of the software),
    - a contract between the software user and the company (securing user's rights, in one side; securing the company against bad use, on the other side).

Trusty institutional agents

- A first and natural attempt would be simply to use the representation notion presented before as a way of transmission of responsibilities.

- But there is a problem: representation is not transitive. We cannot say that:

$$(x:r):REP(y:r1,\phi) \wedge (y:r1):REP(z:r2,\phi) \rightarrow (x:r):REP(z:r2,\phi)$$

  - Representation is a relationship between agents.
  - There might exist a relationship between x and y where it is stated $(x:r):REP(y:r1,\phi)$; there might exit another relationship between y and z where it is stated $(y:r1):REP(z:r2,\phi)$.
  - But from those two relationships we cannot infer that there is a relationship between x and z.

– Example: x may be representative of a company k for ϕ, and the company k may hold the role of single auditor of company i, being representative of i for ϕ. From that we cannot conclude that x is representative of i for ϕ (there is no relationship between them).

Trusty institutional agents

- We will consider (in a very simplistic way) only responsibility for action in a role ($RESP(x : r, \phi)$ means "x acting in role r is responsible for ϕ"):

$$RESP(x : r, \phi) \overset{def}{=} E_{x:r}\phi$$

- If we combine this responsibility concept with the representation concept presented before, we can trace responsibilities for action.

Trusty institutional agents

T-SAR: *a software agent (sa) in a role (r) is trusty for some action (ϕ):*

$$T - SAR(sa : r, \phi) \overset{def}{=} E_{sa:r}\phi \rightarrow \exists_y \exists_{r1}(RESP(y : r1, \phi) \wedge \neg(is - sAg(y)))$$

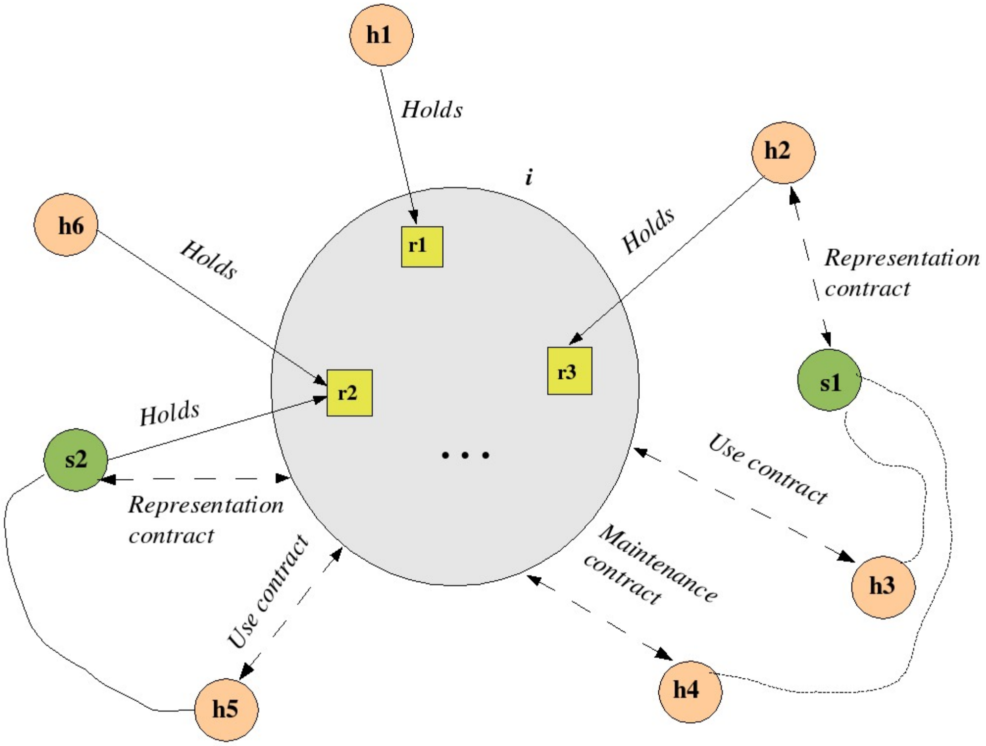
T-SA: *a software agent is trusty for some action:*

$$T - SA(sa, \phi) \overset{def}{=} \forall_r((qual(sa : r) \wedge P_r\phi) \rightarrow T - SAR(sa : r, \phi))$$

T-I: *an institutional agent is trusty for some action:*

$$T - I(i, \phi) \overset{def}{=} \forall_{sa}((is - sAg(sa) \wedge member(sa, i)) \rightarrow T - SA(sa, \phi))$$

An example



Legend:

Human agents: $h_1..h_6$;

Software agents: $s1$ - **railway time-table database,** $s2$ - **ticket machine ;**

Roles: $r1$ - **railway manager,** $r2$ - **ticket seller,** $r3$ - **schedule manager**

An example

$$
\begin{array}{lll}
ST_i & = & < R_i, DCR_i, TO_i, RER_i > \\
R_i & = \{ & is - role(r1, i), \quad is - role(r2, i), \quad is - role(r3, i), \ldots\} \\
DCR_i & = \{ & O_{r1}A1, \quad P_{r1}B1, \\
 & & O_{r2}A2, \quad O_{r2}B2, \\
 & & O_{r3}A3, \ldots\} \\
TO_i & = \{ & O_{i:itself}A1 \rightarrow O_{r1}A1, \\
 & & O_{i:itself}A2 \rightarrow O_{r2}A2, \\
 & & O_{i:itself}A3 \rightarrow O_{r3}A3, \ldots\} \\
RER_i & = \{ & r1 : REP(i : itself, A1), \\
 & & r2 : REP(i : itself, A2), \\
 & & r3 : REP(i : itself, A3), \ldots\}
\end{array}
$$

where:

$A1$ - **Define trains' schedule;**

$B1$ - **Change ticket prices;**

$A2$ - **Collect the appropriate ticket prices;**

$B2$ - **Inform users to use the exact amount of money, when there is no change;**

$A3$ - **Inform about train schedule.**

An example

$$
\begin{array}{lll}
SA & = & < iAg, sAg, hAg, CONT > \\
iAg & = \{ & is - iAg(i)\} \\
sAg & = \{ & is - sAg(s1), is - sAg(s2)\} \\
hAg & = \{ & is - hAg(h1), is - hAg(h2), is - hAg(h3), is - hAg(h4), \\
 & & is - hAg(h5), is - hAg(h6)\} \\
CONT & = \{ & Cont1(h1, i), Cont2(h2, i), Cont3(h6, i), \\
 & & Cont4(s2, i), Cont5(s1, h2) \\
 & & Cont6(h4, i), Cont7(h3, i), Cont8(h5, i)\} \\
Cont1(h1, i) & = & qual(h1 : r1) \\
Cont2(h2, i) & = & qual(h2 : r3) \\
Cont3(h6, i) & = & qual(h6 : r2) \\
Cont4(s2, i) & = & qual(s2 : r2) \wedge (s2 : r2) : REP(i : itself, *) \\
Cont5(s1, h2) & = & qual(s1 : r) \wedge (s1 : r) : REP(h2 : r3, *) \\
Cont6(h4, i) & = & qual(h4 : r4) \wedge qual(i : r5) \wedge O_{i:r5}A4 \wedge O_{h4:r4}B4 \\
Cont7(h3, i) & = & qual(h3 : r6) \wedge O_{h3:r6}A5 \wedge O_{i:itself}B5 \\
Cont8(h5, i) & = & qual(h5 : r8) \wedge O_{h5:r8}A6 \wedge O_{i:itself}B6
\end{array}
$$

Valid actions and fulfillment/unfulfillment of obligations

- Is i a trusty agent?

- Our aim is to verify

  –
  - if an action is valid $E_{a:r}\psi \wedge P_{a:r}\psi$,
  - if there is a fulfillment of some obligation $E_{a:r}\psi \wedge O_{a:r}\psi$ or
  - if there is a violation of some obligation $E_{a:r}\neg\psi \wedge O_{a:r}\psi$.

- $\Delta \vdash_{\mathcal{T}(SA)} \psi$.

  - $\mathcal{T}(SA) \;=\; \mathcal{L}_{\mathcal{DA}} \;+\;$ formulas of SA
  - Δ - a set of action and/or deontic formulas

Chain of responsibilities

$$\Delta \vdash_{\mathcal{T}(SA)} \psi.$$

Case 1: *The railway time-table database s1 gives the user h3 correct information about trains'schedule (A3), which is an obligation of i.*
$\Delta = \{E_{s1:r}A3, O_{i:itself}A3\}$
$\psi = (E_{h2:r3}A3 \wedge O_{h2:r3}A3) \quad \wedge \quad (E_{i:itself}A3 \wedge O_{i:itself}A3)$

Case 2: *The railway time-table database s1 gives the user h3 incorrect information about trains' schedule ($\neg A3$). This failure is due to a technical problem $\neg B4$ of the responsibility of h4 (we will represent this causality by an implication).*
$\Delta = \{E_{h4:r4}\neg B4, E_{h4:r4}\neg B4 \rightarrow E_{s1:r}\neg A3, O_{i:itself}A3\}$
$\psi = (E_{h4:r4}\neg B4 \wedge O_{h4:r4}B4) \quad \wedge \quad (E_{h2:r3}\neg A3 \wedge O_{h2:r3}A3) \quad \wedge$
$(E_{i:itself}\neg A3 \wedge O_{i:itself}A3)$

Future work

- Refine this high-level model:
  - Relate states of affairs with actions.
  - Detail contracts and norms.
- Characterize different levels of responsibility.
- Add dynamics.

Algumas Referências Bibliográficas

- Carmo, J and Pacheco, O.: “Deontic and action logics for organized collective agency, modeled through institutionalized agents and roles”, *Fundamenta Informaticae*, Vol.48 (No. 2,3), pp. 129-163, IOS Press, November, 2001.

- O. Pacheco and J. Carmo: “ A Role Based Model for the Normative Specification of Organized Collective Agency and Agents Interaction”, *Journal of Autonomous Agents and Multi-Agent Systems*, Vol. 6, Issue 2, pp.145-184, Kluwer, March 2003.

- O. Pacheco, “Normative specification: a tool for trust and security”, T.Dimitrakos, F. Martinelli, P. Ryan, S. Schneider (eds.), *Proceedins of the* $3.^{rd}$ *International Workshop on Formal Aspects of Security and Trust* (FAST’05), Newcastle, July, 2005 (to be published by Springer in January 2006).