An agent-based legal knowledge acquisition methodology for design and diagnosis in public administration

Alexander Boer
New results since last meeting

- Agent roles as KA metaphor
- Diagnosis & coordination in agent role-based MAS
- KA as serious gaming
ICAIL 2011 (ACM)
An Agent-based Legal Knowledge Acquisition Methodology for Agile Public Administration
Part I

COMPLAW 2011 & BIS 2011 (Springer LNBIP)
Application of Model-based Diagnosis to Multi-agent Systems representing Public Administration
Part II
Design perspectives towards the organization

- **External perspective:** the organization as an agent or component
- **Internal perspective:** the organization as an environment or system
The knowledge acquisition process

Requirements: from the sources of law

The components: design patterns & design experiences used in the organization

The system: existing employees to educate & resources & structures to preserve or adapt

The environment: experience or evidence-based expectations about the behaviour of agents in the domain
Policy

Environment

Individual case handling

Classify case

Plan

Execute

Monitor

Diagnose

Product & Process development

Model

Design

Implement

Monitor

Diagnose

Sources of Law

Model

Design

Implement

Monitor

Diagnose

Legislation

Implementation knowledge resources
Agent roles as KA metaphor

Alexander Boer
General principle

The sources of law constrain the rules and plans associated to agent roles (the addressees)

By obliging, prohibiting, permitting, & enabling and disabling legal effects

E.g. the buyer knows what an acceptable price for the good is, offers to buy to a seller and monitors acceptance, or accepts an offer to sell from a seller, knows when a sale is concluded, pays the agreed sum, and monitors delivery of the good.
Agent role modeling

Agent role is a self-other(s) representation
A limited perspective on a social system

Prescriptive elements of the agent role:

Abilities
Susceptibilities to action of others
Beliefs, desires, and intentions

Descriptively we distinguish between

Health modes: in conformance with the prescriptive elements of the role

Fault modes: typical abuses of the role known from experience
Agent role modeling: principles

The law addresses persons by agent role

The agent role is the functional component of the social organization

The agent player is a resource scheduled for use by roles

All action, including coordinating action, including player assignment, is performed in a role
Agent role simulation in MAS

Classical MAS paradigm only allows internalization of agent role knowledge (private knowledge)

Extensions for modeling social organization externalize social organization artifacts (common, or at least objective, knowledge)

The MAS paradigm makes a fundamental distinction between internal, single agent, and external, multi-agent, solutions to agent role coordination
Agent role simulation in MAS: design principles

Functional decomposition: each MAS agent represents one agent role

All coordination is external coordination between players, using messages

Agent player identity (when needed) is private knowledge of agent roles

Brute reality is irrelevant: every action is a message sent in a role
Agent role simulation in MAS: why?

**Reason #1:** Flesh-and-blood agent player identity is not always known (criminal law) and not always relevant (commercial law)

**Reason #2:** Law assumes that beliefs, desires, and goals can be localized to agent role (even if same player)

**Reason #3:** In business process design agent players are treated as resources: one employee doing all tasks (case manager) vs. distribution of labour are unproblematic design alternatives
Agent roles as components

Interchangeability of components with the same function

**Design:** Health modes represent alternative functional components with varying requirements on their environment

**Diagnosis:** Health and fault modes are alternative explanations of observations

Law-based, design-based, and experience-based agent roles may all represent health modes or fault modes of the system.
Example: Simple health mode of seller

//seller who sells only for 9, both actively and passively, and doesn't know whether he can deliver

{ include("sale.asl") }

acceptable(sale(seller, Ab, beer, 9)). // 9 is acceptable

!sale(seller, Ab, beer, 9).
Example: Simple fault mode of seller

// seller who sells for 9, except when selling to blacklisted buyers
{ include("sale.asl") }
blacklisted(ihatehim).
acceptable(sale(As, Ab, beer, 11)):- .my_name(As) & blacklisted(Ab).
acceptable(sale(As, Ab, beer, 9)):- .my_name(As) & not blacklisted(Ab).
Example: Another health mode of seller

// zero intelligence plus seller of beer, keeping his own inventory

{ include("sale.asl") }

stock(beer,3).
cost(beer,3).
margin(beer,6).
listprice(Good,Cost+Margin):- cost(Good,Cost) & margin(Good,Margin).

!sell(beer).
+!sell(Good): listprice(Good, ListPrice) <- .my_name(As); !sale(As, Ab, Good, ListPrice); !sell(Good).

acceptable(sale(As, Ab, Good, Sum)) :- stock(Good,Stock) & listprice(Good,ListPrice) & Stock > 0 & Sum >= ListPrice.
+acceptance(As, Ab, sale(As, Ab, Good, Sum)) : cost(Good, Cost) & margin(Good,Margin) & Sum >= (Cost+Margin) & margin > 0 <- -margin(Good,Margin); +margin(Good,Margin+1); .print("Setting list price: ", Margin+1).
+acceptance(As, Ab, sale(As, Ab, Good, Sum))[As] : stock(Good,Stock) & cost(Good, Cost) & margin(Good,Margin) & Sum < (Cost+Margin) & margin > 0 <- -margin(Good,Margin); +margin(Good,Margin-1); .print("Setting list price: ", Margin-1).
+offer(As, Ab, sale(As, Ab, Good, Sum))[As] : stock(Good,Stock) & cost(Good, Cost) & margin(Good,Margin) & Sum < (Cost+Margin) & margin > 0 <- .wait(3000); ?(not acceptance(As, Ab, sale(As, Ab, Good, Sum)))); -margin(Good,Margin); +margin(Good,Margin-1); .print("Setting list price: ", Margin-1).
Background: shared legal knowledge of sales (fragment)

// sale consists of offer and acceptance
sale(As, Ab, Good, Sum) :- offer(As, Ab, sale(As, Ab, Good, Sum)) & acceptance(As, Ab, sale(As, Ab, Good, Sum)).

// to sell is to have a buyer and seller, and to have an offer followed by an accept
+!sale(As, Ab, Good, Sum): seller(As) & buyer(Ab) <- !offer(As, Ab, sale(As, Ab, Good, Sum)); !acceptance(As, Ab, sale(As, Ab, Good, Sum)).

// acceptance must follow an offer to constitute a sale & sale leads to duty to deliver and duty to pay (in no particular order)
+acceptance(As, Ab, sale(As, Ab, Good, Sum)) : sale(As, Ab, Good, Sum) <- !delivery(As, Ab, Good); !payment(As, Ab, Sum).
+acceptance(As, Ab, sale(As, Ab, Good, Sum)) : sale(As, Ab, Good, Sum) <- !payment(As, Ab, Sum); !delivery(As, Ab, Good).

// an offer is made if none exists
+!offer(As, Ab, sale(As, Ab, Good, Sum)) : not offer(As, Ab, sale(As, Ab, Good, Sum)) <- !declare(As, Ab, offer(As, Ab, sale(As, Ab, Good, Sum)));
+offer(As, Ab, sale(As, Ab, Good, Sum)).

// an offer already exists
+!offer(As, Ab, sale(As, Ab, Good, Sum)) : offer(As, Ab, sale(As, Ab, Good, Sum)) <- !declare(As, Ab, offer(As, Ab, sale(As, Ab, Good, Sum)));

// accept an acceptable offer made by another
+!acceptance(As, Ab, sale(As, Ab, Good, Sum)) : offer(As, Ab, sale(As, Ab, Good, Sum))[source(X)] & not myself(X) & not acceptance(As, Ab, sale(As, Ab, Good, Sum)) & acceptable(sale(As, Ab, Good, Sum)) <- !declare(As, Ab, acceptance(As, Ab, sale(As, Ab, Good, Sum)));
+acceptance(As, Ab, sale(As, Ab, Good, Sum)).
Coordinators

- Bank
- Coordinator
- Inventory manager
- Producer
- Seller
- Delivery
- Consumer
- Payments
- Buyer
- Coordinator
Coordinators

- bank
- producer: Don't sell below production costs
- seller: Selling at market prices
- deliver: Don't sell if I can't deliver
- consumer
- inventory manager
- coordinator
- buyer

Sell excess stocks, don't sell items not in stock
Coordinators

Hierarchical: Coordinator orders service production and consumption

Service-oriented (reactive): service consumption by clients drives service production

Market aware: Service production and consumption is adapted to supply and demand monitoring
Example: Simple health mode of seller

//seller who sells only for 9, both actively and passively, and doesn't know whether he can deliver

{ include("sale.asl") } 

acceptable(sale(seller, Ab, beer, 9)). // 9 is acceptable

!sale(seller, Ab, beer, 9).
//seller who sells for 9, except when selling to blacklisted buyers

\{ include("sale.asl") \}
blacklisted(ihatehim).

acceptable(sale(As, Ab, beer, 11)):- .my_name(As) & blacklisted(Ab).
acceptable(sale(As, Ab, beer, 9)):- .my_name(As) & not blacklisted(Ab).
Example: Another health mode of seller

```prolog
// zero intelligence plus seller of beer, keeping his own inventory
{ include("sale.asl") }
stock(beer,3).
cost(beer,3).
margin(beer,6).
listprice(Good,Cost+Margin) :- cost(Good,Cost) & margin(Good,Margin).
!sell(beer).
+!sell(Good) : listprice(Good, ListPrice) <- .my_name(As); !sale(As, Ab, Good, ListPrice); !sell(Good).
acceptable(sale(As, Ab, Good, Sum)) :- stock(Good,Stock) & listprice(Good,ListPrice) & Stock > 0 & Sum >= ListPrice.
+acceptance(As, Ab, sale(As, Ab, Good, Sum)) : cost(Good, Cost) & margin(Good,Margin) & Sum >= (Cost+Margin) & margin > 0 <- -margin(Good,Margin); +margin(Good,Margin+1); .print("Setting list price: ", Margin+1).
+acceptance(As, Ab, sale(As, Ab, Good, Sum))[As] : stock(Good,Stock) & cost(Good, Cost) & margin(Good,Margin) & Sum < (Cost+Margin) & margin > 0 <- -margin(Good,Margin); +margin(Good,Margin-1); .print("Setting list price: ", Margin-1).
+offer(As, Ab, sale(As, Ab, Good, Sum))[As] : stock(Good,Stock) & cost(Good, Cost) & margin(Good,Margin) & Sum < (Cost+Margin) & margin > 0 <- .wait(3000); ?(not acceptance(As, Ab, sale(As, Ab, Good, Sum))); -margin(Good,Margin); +margin(Good,Margin-1); .print("Setting list price: ", Margin-1).
```

Sort of market aware
Diagnosis in agent role-based MAS

Alexander Boer
What does the environment afford & what do I want of it?
How to use the environment effectively to get what I want?

Model → Plan → Execute → Monitor → Diagnose

Model
Plan
Execute
Monitor
Diagnose
Is everything going according to expectations?
What is wrong with my model of the environment?
What does the environment afford & what do I want of it?

Model → Plan → Execute → Monitor → Diagnose
What functionality do I want & which resources do I have?
Is everything according to the design specifications?
Task specialization

Ontvang, kies plan, voer uit

Reactie op impasse in proces

Individual case handling

Model → Design → Implement → Monitor → Diagnose

Product & Process development

Model → Design → Implement → Monitor → Diagnose

Legislation

Sources of Law

Implementation knowledge resources
Internal perspective of the immigration service

I. Receive & Register
II. Collect Information
III. Check
IV. Decide
V. Publish
VI. Hand out

Services

Monitoring & Diagnosis

Enforcement profiles
Experience-based agent profiles

Risk profiles

Enforcement profiles

Legend:
Noncompliance, diagnosis, and enforcement

Motivational Postures

- Disengagement (individual or group has decided not to comply)
- Resistance (individual or group does not want to comply)
- Capitulation (individual or group tries to comply without always succeeding)
- Commitment (individual or group is willing to do the right thing)

Enforcement strategies

- Prosecution
- Audit with/without penalty
- Real time business examinations, record keeping reviews
- Education, record keeping, service delivery (convenience, access, choice, control)

Regulatory strategies

- Command regulation (non-discretionary: use full force of the law)
- Command regulation (discretionary: deter by detection)
- Enforced self-regulation (help to comply)
- Self-regulation (make compliance easy)

Cartoon: "I see you have all of your checks and receipts. What are you trying to hide?"
Noncompliance, diagnosis, and enforcement

Motivational Postures
- Disengagement (individual or group has decided not to comply)
- Resistance (individual or group does not want to comply)
- Capitulation (individual or group tries to comply without always succeeding)
- Commitment (individual or group is willing to do the right thing)

Enforcement strategies
- Prosecution
- Audit with/without penalty
- Real time business examinations, record keeping reviews
- Education, record keeping, service delivery (convenience, access, choice, control)

Regulatory strategies
- Command regulation (non-discretionary: use full force of the law)
- Command regulation (discretionary: deter by detection)
- Enforced self-regulation (help to comply)
- Self-regulation (make compliance easy)

Trust

Deterrence
What is model-based diagnosis?

✓ A diagnostic agent having a model of a system to be diagnosed as its environment.
✓ System can be decomposed into small components with well-understood behaviour models.
✓ Obtain observations on the states implied by the model of the system.
✓ If observations and system description are inconsistent with the assumption that all components are normal, certain of the components behave abnormally.
   ✓ Localize the fault
✓ A diagnosis is a hypothesis that certain of the components are abnormal and the rest normal.
   ✓ Components have health and fault modes.
Multi-agent diagnosis

Objectively, a description of a **social environment** is a pair \((\text{ENV, ROLES})\), where \(\text{ENV}\) is a set of first order sentences, and \(\text{ROLES}\) a set of **agent roles in the environment**.

All \(a\) in \(\text{ROLES}\), are assigned an **agent role class** \(A(a)\) in \(\text{ENV}\), which determines possible **health modes and fault modes**.

The **structural topology** of the system is determined by **messages** \(M(a_1, a_2, m)\), where \(a_1, a_2\) are agent roles and \(m\) is a message, and agents have goals \(G(a, g)\), where \(a\) is an agent role and \(g\) a goal.

A **diagnosis** assigns a health or fault mode to each agent role.
Example: some fault modes of sale

Simple non-payment or non-delivery: these are noncompliance scenarios inherent in the legal description of sale (i.e. The buyer and seller monitor for them)

Also attempts to foil proof standards for completion of the sale

- The payment order is cancelled or the payment is rolled back (storno)
- The delivery is contest by the buyer
- The sale is contested (e.g. By parents of a minor)
Why multi-agent diagnosis?

Agent responsibility assignment as a fault localization problem = model-based diagnosis

Legal knowledge in the form of critical incidents and, for want of a better word, *noncompliance storylines*

1. Have no natural place in traditional KR;
2. Stereotypical intentional agents (e.g. the tax evader, the smuggler) tend to survive policy changes
Issues in multi-agent diagnosis

1. Identification of failing parts of a plan in cooperative business settings
   ✔ Failure caused by differing or false beliefs about the system

2. Localization of agents behaving in a known fault mode
   ✔ Failure caused by intentions

3. Lack of global accessibility of the system for observation
   ✔ Coordinating the process of diagnostic hypothesis generation and testing
Differences with standard model-based diagnosis

A diagnostic agent may behave in a fault mode;
A diagnostic agent does not have access to all messages exchanged between relevant agents; and

1. distinction between messages known first hand, and hypothetical messages of which the agent may receive information from trusted agents
2. Trusted agents may behave in a fault mode;

A diagnostic agent generally has to assume that other agents may reason about behaviour modes and adjust their behaviour accordingly
agents operating in a fault mode may intentionally hide their tracks.
Example setting: real estate crime and taxation

Intentional deviation from real property value:

1. tax evasion
2. bid rigging
3. kickbacks
4. extortion

Key indicators:

1. large deviations from apparent (assessed) property market value, or
2. untypically quick depreciation or appreciation of property value

Small deviations incur a large overhead

overdrachtsbelasting
Real estate crime and taxation

Agents necessarily involved:

1. Buyer
2. Seller
3. Notary lawyer
4. Cadastral registration

Optionally:

1. Real estate agents/advertisers
2. Banks
3. Assessors/appraisers
4. Auction houses
Real estate crime and taxation

1. Distinguishing normal from deviant storylines
2. Judging intentions of buyer and seller
3. Setting monitoring processes in place
4. Finding sources of evidence of coordination between parties, or the possibility of coordination
   - Hypothesized coordinators
   - Internal and external coordination
Real estate crime and taxation
Simple fault mode of seller

//seller who sells for 9, except when selling to blacklisted buyers
{ include("sale.asl") }
blacklisted(ihatehim).
acceptable(sale(As, Ab, beer, 11)):- .my_name(As) & blacklisted(Ab).
acceptable(sale(As, Ab, beer, 9)):- .my_name(As) & not blacklisted(Ab).
Real estate crime and taxation: typical optionals

- Coordinator
- Buyer
- Seller
- Assessor
- Auction house

Creates a closed market where all bidders are known.

Determines price to add/subtract kickback to/from.
Real estate crime: seller kickback plan (abstract)

! transfer addv to recipient
+! transfer addv to recipient <- propose to sell a property worth mv for mv - addv.
+ Plan to sell a property worth mv for mv - addv accepted <- propose a property worth mv.
+ Property is worth mv is accepted ! <- secure property for mv; ! sell property for mv - addv.
Flaws and repairs in the simple kickback scheme

Acquiring the property and then immediately selling it for a loss & then again for a profit is suspicious and obvious in Cadastral data

✓ Chain or aggregate transactions to hide losses and profits
✓ More accomplices increases risk of information leaks
✓ Accomplices may extort money
  ✓ Sign all deeds at the same time in the presence of a notary lawyer
    ✓ Notary lawyer may report a suspicious transaction
      ✓ Make notary lawyer an accomplice
        ✓ Notary lawyer may report real estate fraud
        ✓ Notary lawyer must be paid
Flaws and repairs in the simple kickback scheme

Consulting an assessor and then intentionally deviating from assessed value is suspicious to the assessor

✓ Make assessor an accomplice
  ✓ Assessor may report real estate fraud
  ✓ Assessor must be paid
Conclusions

KA games make knowledge fit the framework
One model for internal and external coordination

The proposed diagnosis framework is an complementary approach to
- causal analysis of noncompliance stories and
- argumentation schemes about noncompliance and burden of proof
KA as serious gaming

Alexander Boer
In interviews, valuable knowledge often comes in the form of anecdotes.

Structured KA makes knowledge fit more readily into a knowledge framework [8].

Setting up KA games motivates development teams to make the most of KA meetings [9].

KA game test run at the immigration service.
Knowledge acquisition games

Participants from design, work floor, enforcement, policy making

Participants take roles, and are encouraged to play out anecdotes

Designers teach back & formalize as agent role descriptions what they learn
Focus on:

Anticipated noncompliance scenarios
Sources of law accessed in service delivery
Content of messages exchanged
Evidence asked for
Record keeping and scoring in games

How to reward skilled participation?

Records directly kept in simplified AgentSpeak

- Practical problem: when to open a new agent role?
- Ambiguity in interpretation: initiative vs. waiting for events
Conclusions

MAS paradigm is most useful for a KA games framework if we understand agents as agent roles.

Agent roles are self-other representations.

The proposed framework can support both model-based design and diagnosis in the problem solving cycle.

The proposed framework accounts for critical incident/noncompliance stories.
Next steps

Inventory of design patterns for
  – Service and sale
  – Other models of resource allocation
  – Obligation, prohibition, permission
  – Power/liability, disability/immunity

A first serious gaming environment

More serious game testing