Λογική Επιχειρηματολογίας
(Προγραμματισμός Κοινής Λογικής)

Software Agents &
Cognitive Systems

via

Argumentation

An Example of Argumentation Decision Policy

- **Decision policy of a seller agent**

- Normally, sell a product at its high price.

You can sell a product at the lower price only if payment is cash. (But normally prefer to sell high.)

Regular customers can be offered the low price.*

In high season you must sell at high prices.

* This could be conditional e.g. to buy 2 items, etc.

- **Options: sell_high or sell_low**

Seller agent: Scenarios

- <1, {}, sell(Prd,Ag, high)>
- <2, {pay_cash(Prd,Ag)}, sell(Prd,Ag, high)>
- <3, {pay_cash(Prd,Ag), regular(Ag)}, sell(Prd,Ag, high); sell(Prd,Ag, low)>
  - Non-deterministic Scenario
- <4, {pay_cash(Prd,Ag), regular(Ag), high_season}, sell(Prd,Ag, high)>
- <5, {pay_cash(Prd,Ag), regular(Ag), not high_season}, sell(Prd,Ag, low)>

Decision policy: seller agent

- Object-level argument rules:

  r1: sell(Prd, Ag, high) ← true
  r2: sell(Prd, Ag, low) ← pay-cash(Ag, Prd)

- Default Priority: r1 > r2

- We also need to express prices are contrary
  - r3: ¬sell(Prd, Ag, P2) ← sell(Prd, Ag, P1), P2≠P1
  - Complementary relation:
    - complement(sell(Prd, Ag, high) , sell(Prd, Ag, low)).

Decision policy: seller agent

- Object-level argument rules:

  r1(Prd,Ag): sell(Prd, Ag, high) ← true
  r2(Prd,Ag): sell(Prd, Ag, low) ← pay-cash(Ag, Prd)

- Priority rules:
  - Generally, sell at high prices:

    R1(Prd,Ag): h-p(r1(Prd,Ag), r2(Prd,Ag)) ← true
  - Regular customers can have low price:

    R2(Prd,Ag): h-p(r2(Prd,Ag), r1(Prd,Ag)) ← regular(Ag)
  - But not at high season:

    C1(Prd,Ag): h-p(R1(Prd, Ag), R2(Prd, Ag)) ← high-season
    C2(Prd,Ag): h-p(R2(Prd, Ag), R1(Prd, Ag)) ← not high-season

Seller agent: Structure of Policy

- Default Policy: "Sell high"
  - For normal markets and normal customers
- Exceptional Policy: "Sell low"
  - For special markets and customers, e.g. regular customers
- Generally, Exceptional (or Special) policies dominate over the Default (or Normal) ones.
  - For normal exceptional cases, i.e. normal market
  - This is a Meta-Default policy!
- Exceptional Policy over the special policy:
  - Exceptional context of high season market.

Seller agent: Argumentation in Scenarios

- <1, {}, sell(Prd,Ag, high)>
  - Only A={r1(p, ag)} applicable argument: supports option high.
  - Hence A is only **admissible** argument.
  - Hence **sceptical decision**: to sell high.

- <2, {pay_cash(Prd,Ag)}, sell(Prd,Ag, high)>
  - A={r1(p, ag)} supports option high price .
  - B={r2(p, ag)} supports contrary option of low price.
    - A **attacks** B and vice-versa
  - A'={r1(p, ag), R1(p,ag)} strengthens A
    - A' **attacks** B but B does **not attack** A'
  - Also B cannot be strengthened (by any applicable priority rule)
  - Hence B cannot be made admissible
  - Hence **sceptical decision**: to sell high.

Seller agent: Argumentation in Scenarios

- <3, {pay_cash(Prd,Ag), regular(Ag)}, sell(Prd,Ag, high); sell(Prd,Ag, low)> - Non-deterministic Scenario
  - B'={r2(p, ag), R2(p,ag)} strengthens B
    - A' **attacks** B' **and** B' **attacks** A'
  - Both A' and B' are admissible.
  - Hence both high and low are credulous conclusions/decisions.

- A' and B' are in conflict not only on the price but also on the priority of r1(…) over r2(…):
  - They conflict on L= h_p(r1(Prd,Ag), r2(Prd,Ag))
  - They argue about the priority or strength of rules.

Seller agent: Argumentation in Scenarios

A' and B' are in conflict not only on the price but also on the decision of priority: h_p(r1(Prd,Ag), r2(Prd,Ag)).

- <4, {pay_cash(Prd,Ag), regular(Ag), high_season}, sell(Prd,Ag, high)>
  - A''={r1(p, ag), R1(p,ag), C1(p,ag)} strengthens A' (and A)
  - A'' attacks B' but not vice versa (on h_p(r1(Prd,Ag), r2(Prd,Ag)))
  - A'' admissible – No admissible argument for low.
  - Hence sceptical decision of high price
- <5, {pay_cash(Prd,Ag), regular(Ag), not high_season}, sell(Prd,Ag, low)>
  - B''={r2(p, ag), R2(p,ag), C2(p,ag)} strengthens B' (and B)
  - B'' attacks A' but not vice versa.
  - Hence sceptical decision of low.

Decision Making in Argumentation

Cognitive Call Assistant

- **Options: allow(call), deny(call)**
- **Preferences: According to User values**
- **General, Cognitive Form of Preferences:**
  - "Generally, in **SITUATION** prefer Oi,
    but when in particular **CONTEXT**, prefer Oj."
  - **"Generally, deny calls when {busy at work}
    but allow calls from {collaborators}."**

Cognitive Call Assistant

- **Decision policy of call assistant:**
  - Normally, allow calls.

    When at work deny calls from unknown numbers.When in a meeting at work also deny known calls unless family calls when there is an emergency at home. Allow all calls from my manager.

- **Options: allow(call), deny(call)**

Call Assistant: Scenarios

- <1, {} , allow(call)>
- <2, {unknown(call),at_work}, deny(call)>
- <3, {in_meeting, at_work } , deny(call)>
- <4, {in_meeting, at_work, family(call),emergency} , allow(call)>
- <5, {in_meeting, at_work ,manager(call)}, allow(call)>

Call Assistant: Extra Scenarios

- <11, {unknown(call) } , allow(call)>
- <44, {in_meeting,at_work,family(call)}, deny(call)>
- <55, { manager(call) } , allow(call)>

Assistant Policy (1)

- Object-level argument rules:

  r1(Call): allow(Call) ← true
  r2(Call): deny(Call) ← true

- Default Priority:
  - Generally, allow calls:
  - R1(Call): r1(Call) > r2(Call) ← true

- Special – Contextual- Priority:
  - Generally, deny unknown calls when at work:
  - R2(Call): r2(Call) > r1(Call) ← unknown(Call), at_work
  - C2(Call): R2(Call) > R1(Call) ← true

Assistant Policy (2)

- Special Contextual Priority:
  - Generally, deny calls when at a work meeting:
  - R4(Call): r2(Call) > r1(Call) ← at_work, in_meeting
  - C4(Call): R4(Call) > R1(Call)) ← true

  - 1. Except, when a family call
  - C1(Call): R1(Call) > R4(Call)) ← family(Call)
  - D1(Call): C1(Call) > C4(Call)) ← true

  - 2. Except, when a family call and emergency
  - C1(Call): R1(Call) > R4(Call)) ← family(Call), emergency
  - D1(Call): C1(Call) > C4(Call)) ← true

Assistant Policy (3)

- Default Priority:
  - Generally, allow calls:
  - R1(Call): r1(Call) > r2(Call) ← true

- Generally, allow calls from manager:
  - This is like a new default priority/policy
  - R3(Call): r1(Call) > r2(Call) ← manager(Call)
  - What higher order priorities, if any, are needed for R3?
    - Priority of manager calls is global – another policy thread
  - New object-level argument rule:

    r3(Call): allow(Call) ← manager(Call)
    R31(Call): r3(Call) > r2(Call) ← true

Call Assistant: Argumentation in Scenarios

- <1, {}, allow(Call)>
  - A={r1(call)} argument supports option allow.
  - B={r2(call)} argument supports option deny.
  - A attacks B and vice versa.
  - A'={r1(call), R1(call)} strengthens A
    - A' **attacks** B but B does **not attack** A'
  - Also B cannot be strengthened (by any applicable priority rule)
  - Hence B cannot be made admissible
  - Hence **sceptical decision**: allow the call.