Formalisation of Dynamic Properties of Multi-Issue Negotiation

Bosse, T.; Jonker, C.; Treur, J.

2006

document version
Publisher's PDF, also known as Version of record

Formalisation of Dynamic Properties of Multi-Issue Negotiation

Tibor Bosse[1], Catholijn M. Jonker[1], and Jan Treur[1,2]

[1] Vrije Universiteit Amsterdam, Department of Artificial Intelligence, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
{tbosse, jonker, treur}@cs.vu.nl
http://www.cs.vu.nl/~{tbosse, jonker, treur}
[2] Utrecht University, Department of Philosophy, Heidelberglaan 8, 3584 CS Utrecht, The Netherlands

bid_alternation(γ:trace)
Over time the bids of A and B alternate: thus for all two different moments in time t1, t3, that A generated a bid, there is a moment in time t2, with t1 < t2 < t3, such that A received a bid generated by B.
∀ A, B: AGENT, ∀ b1, b3: BID, ∀ t1, t3:
t1 < t3 &
state(γ, t1, output(A)) |== to_be_communicated_to_by(b1, B, A) &
state(γ, t3, output(A)) |== to_be_communicated_to_by(b3, B, A) ⇒
∃b2, ∃t2: t1 < t2 < t3 &
state(γ, t2, input(A)) |== communicated_to_by(b2, A, B)

is_followed_by(γ:trace, A:AGENT, t1:time, b1:BID, B:AGENT, t2:time, b2:BID)
In a negotiation process γ bid b1 at time t1 is followed by a bid b2 at time t2 iff bids b1 and b2 are subsequent bids in γ.
state(γ, t1, output(A)) |== to_be_communicated_to_by(b1, A, B) &
state(γ, t2, output(B)) |== to_be_communicated_to_by(b2, B, A) &
t1 < t2 &
[∀t3, ∀C, D: AGENT, ∀b3: BID:
t1 < t3 < t2 ⇒ state(γ, t3, output(C)) |=/= to_be_communicated_to_by(b3, C, D)]

agent_consecutively_bids_to(γ:trace, A:AGENT, t1:time, b1:BID, t2:time, b2:BID, B:AGENT)
In a negotiation process γ agent A consecutively bids b1 at time t1 and then b2 at time t2 to agent B.
state(γ, t1, output(A)) |== to_be_communicated_to_by(b1, A, B) &
state(γ, t2, output(A)) |== to_be_communicated_to_by(b2, A, B) &
t1 < t2 &
[∀t3, ∀b3: BID:
t1 < t3 < t2 ⇒ state(γ, t3, output(A)) |=/= to_be_communicated_to_by(b3, A, B)]

stop_criterion(γ:trace, A:AGENT, t2:time)
The stop criterion holds for agent A at time t, if at time t agent A receives a bid by negotiation partner B that is at least as good as the last bid made by A.
∃t1, ∃B: AGENT, ∃b1, b2: BID:
state(γ, t2, input(A)) |== communicated_to_by(b2, A, B) &
state(γ, t1, output(A)) |== to_be_communicated_to_by(b1, B, A) &
is_followed_by(γ, t1, b1, t2, b2) &
util(γ, A, b1) ≤ util(γ, A, b2)

negotiation_continuation(γ:trace)
For both A and B, unless the stop criterion holds, a new proposal is generated by A upon receival of a proposal by B.
∀t, ∀A, B: AGENT, ∀b1: BID:
¬stop_criterion(γ, A, t) &
state(γ, t, input(A)) |== communicated_to_by(b1, A, B) ⇒
[∃b2: BID ∃t2: t2 > t & state(γ, t2, output(A)) |== to_be_communicated_to_by(b2, B, A)]

strictly_dominates(b1:BID, b2:BID, A:AGENT, B:AGENT)
A bid b1 dominates a bid b2 with respect to agents A and B iff both agents prefer bid b1 over bid b2.
∀vA1, vA2, vB1, vB2 : real :
util(A, b1, vA1) & util(A, b2, vA2) & util(B, b1, vB1) & util(B, b2, vB2) ⇒
vA1 > vA2 & vB1 > vB2

weakly_dominates(b1:BID, b2:BID, A:AGENT, B:AGENT)
A bid b1 dominates a bid b2 with respect to agents A and B iff both agents prefer bid b1 over bid b2.
∀vA1, vA2, vB1, vB2 : real :
util(A, b1, vA1) & util(A, b2, vA2) & util(B, b1, vB1) & util(B, b2, vB2) ⇒
vA1 ≥ vA2 & vB1 ≥ vB2

strictly_better_social_welfare(b1:BID, b2:BID, A:AGENT, B:AGENT)
The social welfare of bid b1 is better than that of bid b2 with respect to agents A and B iff the sum of the utility values of bid b1 is bigger than the sum of the utility values of bid b2. See also [6,10].
∀vA1, vA2, vB1, vB2 : real :
util(A, b1, vA1) & util(A, b2, vA2) & util(B, b1, vB1) & util(B, b2, vB2) ⇒
vA1 + vB1 > vA2 + vB2

strictly_better_equitability(b1:BID, b2:BID, A:AGENT, B:AGENT)
A bid b1 has a better equitability than bid b2 with respect to agents A and B iff the difference in the utility values of bid b1 is less than the difference in utility values of bid b2.
∀vA1, vA2, vB1, vB2 : real :
util(A, b1, vA1) & util(A, b2, vA2) & util(B, b1, vB1) & util(B, b2, vB2) ⇒
| vA1 - vB1 | < | vA2 - vB2 |

ε-equitability(b:BID, A:AGENT, B:AGENT, ε:real)
A bid b has ε-equitability with respect to agents A and B iff the difference in the utility values of bid b is less than ε. Thus, a bid that has an equitability of 0 has a maximum equitability. This definition corresponds to the idea of Raiffa to maximize the minimum utility [10].
∀vA, vB : real :
util(A, b, vA) & util(B, b, vB) ⇒
| vA - vB | ≤ ε

pareto_inefficiency(b:BID, A:AGENT, B:AGENT, ε:real)
With respect to agents A and B, the Pareto inefficiency of a bid b is the number ε that indicates the distance to the Pareto Efficient Frontier according to some distance measure d in utilities. Here d(b1, b2) is the distance between the bids b1 and b2 when viewed as points in the plane of utilities.
∀vA, vB : real :
util(A, b, vA) & util(B, b, vB) ⇒
pareto_distance(vA, vB) = ε

making_global_concession(γ:trace, A:AGENT, t1:time, b1:BID, t2:time, b2:BID, B:AGENT)

In a negotiation process γ agent B makes a global concession to agent B with respect to bid b1 at time t1 and bid b2 at time t2 iff both bids are consecutive, and b2 has a lower utility than b1, from A's perspective. A similar property could be defined stating that an agent receives a global concession from another agent.

agent_consecutively_bids_to(γ, A, t1, b1, t2, b2, B) &
∀vA1, vA2 : real :
util(A, b1, vA1) & util(A, b2, vA2) ⇒
vA1 > vA2

configuration_differs(b1:BID, b2:BID)

Two bids b1 and b2 differ in configuration iff there is an issue that has a different value in both bids. Similar properties could be defined stating that two bids differ in configuration in at least x issues.

∃a: ISSUE, ∃v1, v2: VALUE:
value_of(b1, a, v1) &
value_of(b2, a, v2) &
v1 ≠ v2

agent_views_agent_makes_config_variation(γ:trace, A:AGENT, B:AGENT, t1:time, b1:BID, t2:time, b2:BID)

In the view of agent A, agent B varies the configuration, but not the utility. Note that one agent can both be agent A and B, or A and B can refer to different agents.

agent_consecutively_bids_to(γ, A, t1, b1, t2, b2, B) &
configuration_differs(b1, b2) &
∀vA1, vA2 : real :
util(A, b1, vA1) & util(A, b2, vA2) ⇒
vA1 = vA2

agent_views_agent_makes_strict_ε-progression(γ:trace, A:AGENT, B:AGENT, t1:time, b1:BID, t2:time, b2:BID, ε:real)

In the view of agent A, the two consecutive bids b1 and b2 made at times t1 and t2 by agent B show minimum ε-progression in utility iff the second bid is at least ε higher than the first bid. Note that one agent can both be agent A and B, or A and B can refer to different agents.

agent_consecutively_bids_to(γ, A, t1, b1, t2, b2, B) &
∀vA1, vA2 : real :
util(A, b1, vA1) & util(A, b2, vA2) ⇒
vA2 - vA1 > ε

strict_pareto_monotony(γ:trace, tb:time, te:time)

A negotiation process γ is Strictly Pareto-monotonous for the interval [t1, t2] iff for all subsequent bids b1, b2 in the interval b2 dominates b1:

∀t1, t2, ∀A, B: AGENT, ∀b1, b2: BID
[tb ≤ t1 < t2 ≤ te & is_followed_by(γ, A, t1, b1, B, t2, b2)]
⇒ strictly_dominates(γ, b2, b1, A, B)

weak_pareto_monotony(γ:trace, tb:time, te:time)

A negotiation process γ is Weakly Pareto-monotonous for the interval [t1, t2] iff for all subsequent bids b1, b2 in the interval b2 weakly dominates b1:

∀t1, t2, ∀A, B: AGENT, ∀b1, b2: BID
[tb ≤ t1 < t2 ≤ te & is_followed_by(γ, A, t1, b1, B, t2, b2)]
⇒ weakly_dominates(γ, b2, b1, A, B)