

Automated Negotiation: Challenges and Tools

Yasser Mohammad^{1,2,3} and Amy Greenwald⁴

¹ NEC CORPORATION, Global Innovation Unit

² National Institute of Advanced Industrial Science and Technology (AIST)

³ Assiut University, Egypt

⁴ Brown University

February 23rd, 2022

Outline

1 Why?

2 What?

Why Now?

- ① Industries are moving online.
- ② Automation: Factory floor
The back office.



Outline

- ① Introduction and Classic Results (45min)
- ② Protocols, Strategies and Platforms (45min)
 - ① Hands On Experience
- ③ Recent Advances (45min):
- ④ Supply Chain Management Competition (30min)
 - ① Hands On Experience
- ⑤ Challenges and Open Problems (30min)

Materials

- ① Tutorial Website: http://yasserm.com/aaai2022tutorial-automated_negotiation_challenges_and_tools/
- ② Github Repository:

Why?



Q&A

What?



NEC-AIST
AI Cooperative
Research Laboratory

Automated Negotiation: Challenges and Tools

Introduction and Classic Results

Yasser Mohammad^{1,2,3} and Amy Greenwald⁴

¹ NEC CORPORATION, Global Innovation Unit

² National Institute of Advanced Industrial Science and Technology (AIST)

³ Assiut University, Egypt

⁴ Brown University

February 23rd, 2022

Outline

1 Negotiation

2 Classic Results

3 References

Outline

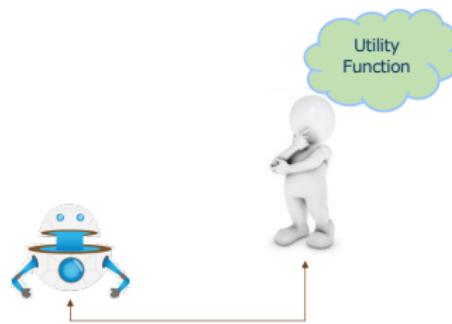
1 Negotiation

2 Classic Results

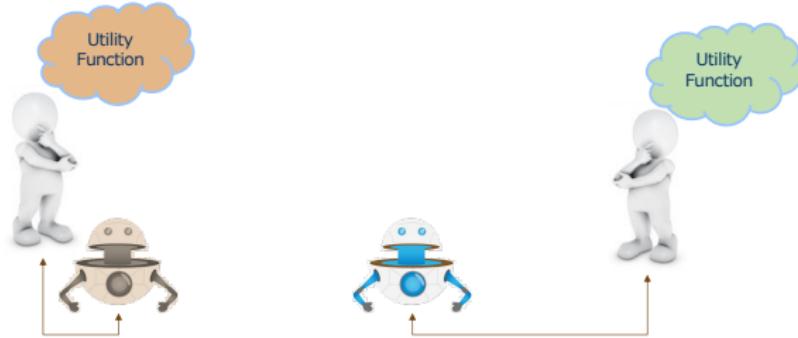
3 References



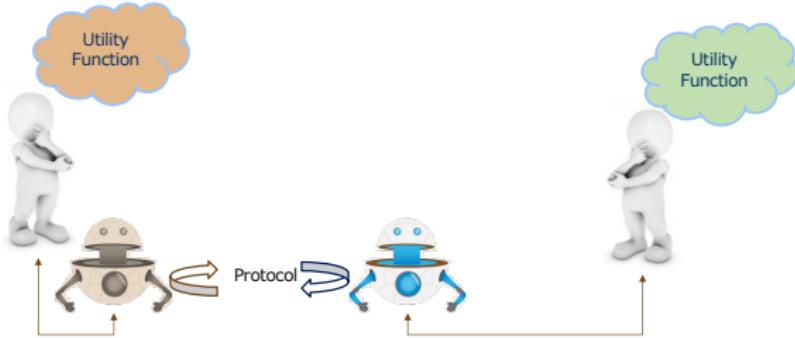
- A method to achieve agreement among self-interested actors.
- Negotiation is important → win-win agreements.
- Automatic Negotiation → \$\$\$
 - smart contracts, resource allocation, SCM, etc



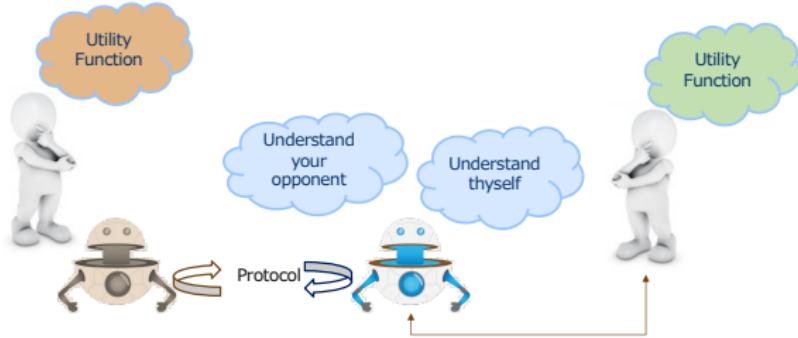
- A method to achieve agreement among self-interested actors.
- Negotiation is important → win-win agreements.
- Automatic Negotiation → \$\$\$
 - smart contracts, resource allocation, SCM, etc



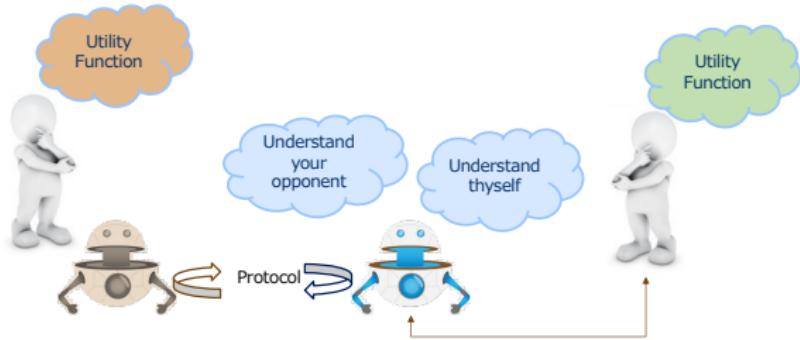
- A method to achieve agreement among self-interested actors.
- Negotiation is important → win-win agreements.
- Automatic Negotiation → \$\$\$
 - smart contracts, resource allocation, SCM, etc



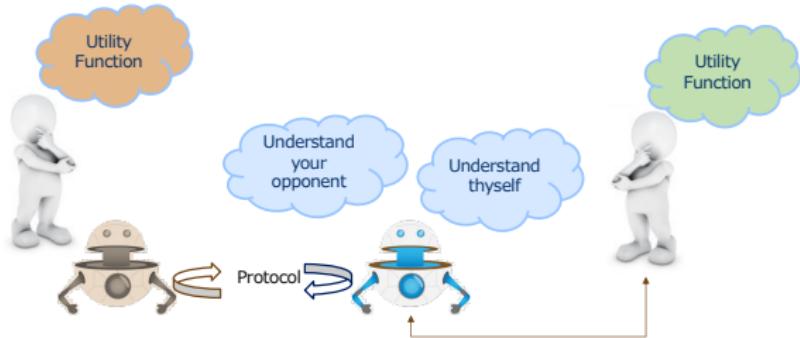
- A method to achieve agreement among self-interested actors.
- Negotiation is important → win-win agreements.
- Automatic Negotiation → \$\$\$
 - smart contracts, resource allocation, SCM, etc



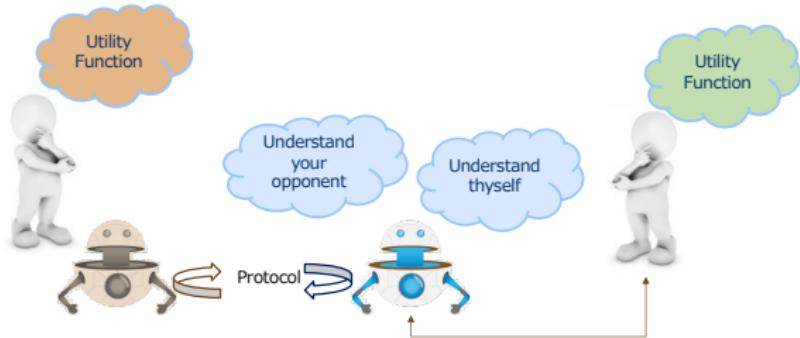
- A method to achieve agreement among self-interested actors.
- Negotiation is important → win-win agreements.
- Automatic Negotiation → \$\$\$
 - smart contracts, resource allocation, SCM, etc



- A method to achieve agreement among self-interested actors.
- Negotiation is important → win-win agreements.
- Automatic Negotiation → \$\$\$
 - smart contracts, resource allocation, SCM, etc



- A method to achieve agreement among self-interested actors.
- Negotiation is important → win-win agreements.
- Automatic Negotiation → \$\$\$
 - smart contracts, resource allocation, SCM, etc



- A method to achieve agreement among self-interested actors.
- Negotiation is important → win-win agreements.
- Automatic Negotiation → \$\$\$
 - smart contracts, resource allocation, SCM, etc

Why not Auctions

or when to not use automated negotiation

Definition

Negotiation

$$\mathcal{Y} \equiv \left(A, T, N, \Omega, M, \left\{ \tilde{P}_a, P_{ab}^n \mid 1 \leq a, b \leq A, 0 \leq n \leq N \right\} \right)$$

Definition

Negotiation

$$\mathcal{Y} \equiv \left(A, T, N, \Omega, M, \left\{ \tilde{P}_a, P_{ab}^n \mid 1 \leq a, b \leq A, 0 \leq n \leq N \right\} \right)$$

$A \in \mathbb{I}^+ - \{1\}$: Number of agents/actors.

Definition

Negotiation

$$\mathcal{Y} \equiv \left(A, T, N, \Omega, M, \left\{ \tilde{P}_a, P_{ab}^n \mid 1 \leq a, b \leq A, 0 \leq n \leq N \right\} \right)$$

$A \in \mathbb{I}^+ - \{1\}$: Number of agents/actors.

$T \in \mathbb{R} \cup \infty$: The allowed time of the negotiation.

Definition

Negotiation

$$\mathcal{Y} \equiv \left(A, T, N, \Omega, M, \left\{ \tilde{P}_a, P_{ab}^n \mid 1 \leq a, b \leq A, 0 \leq n \leq N \right\} \right)$$

$A \in \mathbb{I}^+ - \{1\}$: Number of agents/actors.

$T \in \mathbb{R} \cup \infty$: The allowed time of the negotiation.

$N \in \mathbb{I} \cup \infty$: The allowed number of rounds of the negotiation.

Definition

Negotiation

$$\mathcal{Y} \equiv \left(A, T, N, \Omega, M, \left\{ \tilde{P}_a, P_{ab}^n \mid 1 \leq a, b \leq A, 0 \leq n \leq N \right\} \right)$$

$A \in \mathbb{I}^+ - \{1\}$: Number of agents/actors.

$T \in \mathbb{R} \cup \infty$: The allowed time of the negotiation.

$N \in \mathbb{I} \cup \infty$: The allowed number of rounds of the negotiation.

$\Omega \equiv \{\omega_j\} \cup \phi$: Possible outcomes including ϕ signifying disagreement.

Definition

Negotiation

$$\mathcal{Y} \equiv \left(A, T, N, \Omega, M, \left\{ \tilde{P}_a, P_{ab}^n \mid 1 \leq a, b \leq A, 0 \leq n \leq N \right\} \right)$$

$A \in \mathbb{I}^+ - \{1\}$: Number of agents/actors.

$T \in \mathbb{R} \cup \infty$: The allowed time of the negotiation.

$N \in \mathbb{I} \cup \infty$: The allowed number of rounds of the negotiation.

$\Omega \equiv \{\omega_j\} \cup \phi$: Possible outcomes including ϕ signifying disagreement.

M The negotiation *mechanism* (protocol) defining rules of encounter for agents.

Definition

Negotiation

$$\mathcal{Y} \equiv \left(A, T, N, \Omega, M, \left\{ \tilde{P}_a, P_{ab}^n \mid 1 \leq a, b \leq A, 0 \leq n \leq N \right\} \right)$$

$A \in \mathbb{I}^+ - \{1\}$: Number of agents/actors.

$T \in \mathbb{R} \cup \infty$: The allowed time of the negotiation.

$N \in \mathbb{I} \cup \infty$: The allowed number of rounds of the negotiation.

$\Omega \equiv \{\omega_j\} \cup \phi$: Possible outcomes including ϕ signifying disagreement.

M The negotiation *mechanism* (protocol) defining rules of encounter for agents.

\tilde{P}_a : Preferences of **actor** a .

Definition

Negotiation

$$\mathcal{Y} \equiv \left(A, T, N, \Omega, M, \left\{ \tilde{P}_a, P_{ab}^n \mid 1 \leq a, b \leq A, 0 \leq n \leq N \right\} \right)$$

$A \in \mathbb{I}^+ - \{1\}$: Number of agents/actors.

$T \in \mathbb{R} \cup \infty$: The allowed time of the negotiation.

$N \in \mathbb{I} \cup \infty$: The allowed number of rounds of the negotiation.

$\Omega \equiv \{\omega_j\} \cup \phi$: Possible outcomes including ϕ signifying disagreement.

M The negotiation *mechanism* (protocol) defining rules of encounter for agents.

\tilde{P}_a : Preferences of **actor** a .

P_{ab}^n : Information available to **agent** a about preferences of **actor** b at the beginning of round n .

Definition

Negotiation

$$\mathcal{Y} \equiv \left(A, T, N, \Omega, M, \left\{ \tilde{P}_a, P_{ab}^n \mid 1 \leq a, b \leq A, 0 \leq n \leq N \right\} \right)$$

$A \in \mathbb{I}^+ - \{1\}$: Number of agents/actors.

$T \in \mathbb{R} \cup \infty$: The allowed time of the negotiation.

$N \in \mathbb{I} \cup \infty$: The allowed number of rounds of the negotiation.

$\Omega \equiv \{\omega_j\} \cup \phi$: Possible outcomes including ϕ signifying disagreement.

M The negotiation *mechanism* (protocol) defining rules of encounter for agents.

\tilde{P}_a : Preferences of **actor** a .

P_{ab}^n : Information available to **agent** a about preferences of **actor** b at the beginning of round n .

Known Preferences Assumption

$$P_{aa}^n = \tilde{P}_a$$

Preference Representations

Preference Types

Partial Ordering \succsim Defines preference as a partial ordering over Ω .

Ranking A total ordering over a subset of Ω .

Utility Function \tilde{u} Defines a numeric value for every outcome in Ω .

$$\tilde{u} : \Omega \rightarrow \mathbb{R}$$

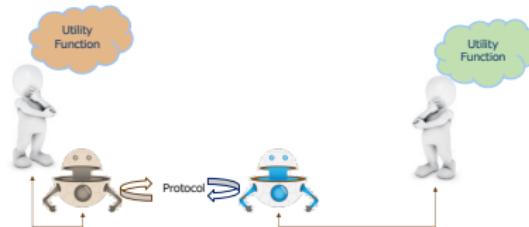
Probabilistic Utility Function u Defines a distribution of values.

$$u : \Omega \times \mathbb{R} \rightarrow [0, 1]$$

Known Ufun Assumption

$$u_a^t(\omega, x) = u_a^0(\omega, x) = \begin{cases} 1 & \tilde{u}(\omega) = x \\ 0 & \text{otherwise} \end{cases}$$

Components of the Negotiation Problem



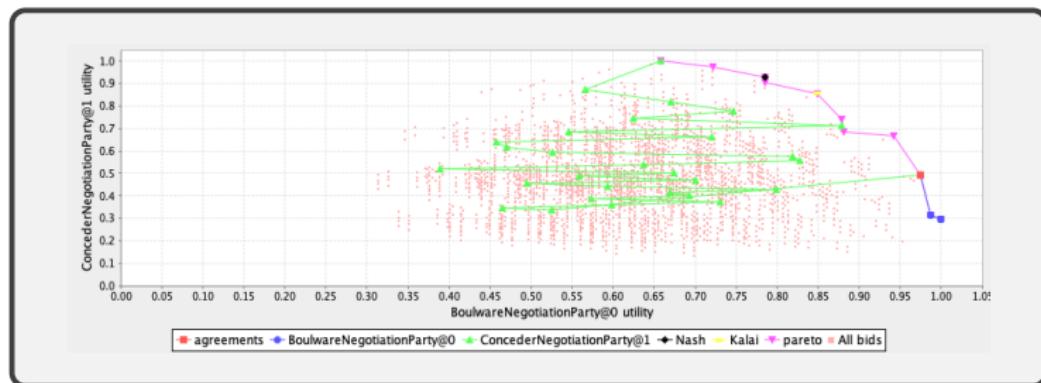
Negotiation Protocol Defines how negotiation is to be conducted [Mechanism Design Problem].

- Alternating Offers Protocol
- Single Text Protocol
- ...

Negotiation Strategy Defines how an agent behaves during the negotiation [Effective Negotiation Problem].

- Time-based strategies: Boulware, conceder, ...
- Tit-for-tat variations
- ...

Important Concepts



Pareto Frontier Outcomes that cannot be improved for one actor without making another worse off.

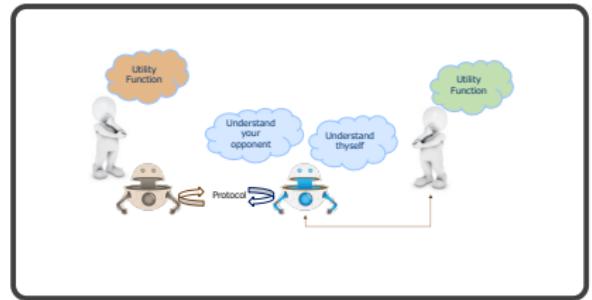
Welfare Total utility received by all actors.

Surplus utility Utility above disagreement utility.

Nash Equilibrium Strategies that are best responses to each other.

Sub-game Perfect Equilibrium A Nash Equilibrium in every sub-game.

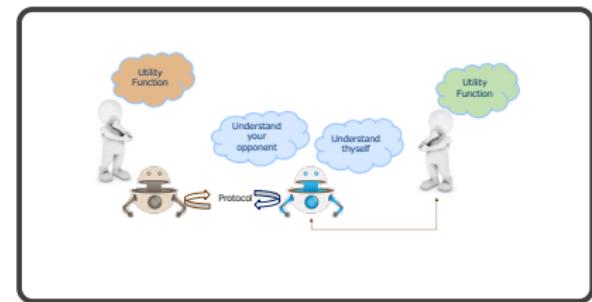
Types of Automated Negotiation Problems



Types of Automated Negotiation Problems

Negotiator type

- ① Agent-Agent negotiation
- ② Agent-Human negotiation



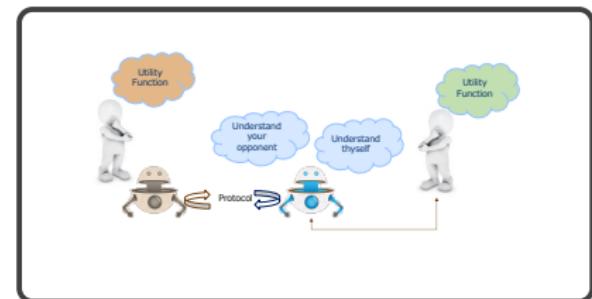
Types of Automated Negotiation Problems

Negotiator type

- ① Agent-Agent negotiation
- ② Agent-Human negotiation

Number of negotiators

- ① Bilateral negotiation
- ② Multilateral negotiation



Types of Automated Negotiation Problems

Negotiator type

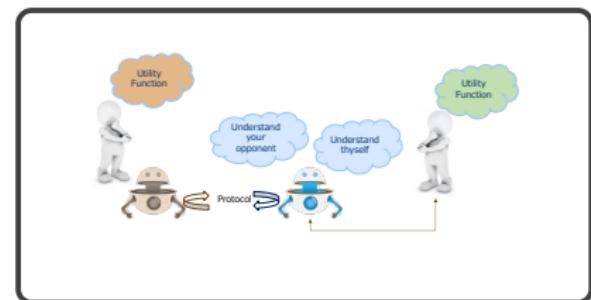
- ① Agent-Agent negotiation
- ② Agent-Human negotiation

Number of negotiators

- ① Bilateral negotiation
- ② Multilateral negotiation

Outcome Space

- ① Single Issue: $\Omega = \{\omega_0, \omega_1, \dots\}$
- ② Multiple Issues: $\Omega = \prod_{i=1}^{n_i} I_i$



Types of Automated Negotiation Problems

Negotiator type

- ① Agent-Agent negotiation
- ② Agent-Human negotiation

Number of negotiators

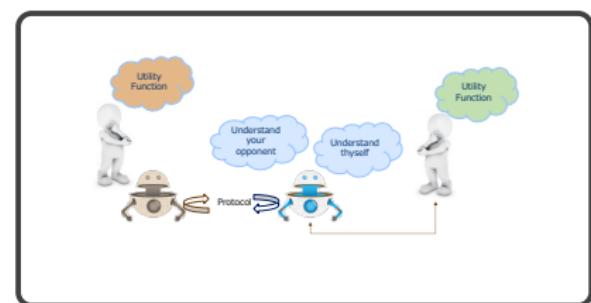
- ① Bilateral negotiation
- ② Multilateral negotiation

Outcome Space

- ① Single Issue: $\Omega = \{\omega_0, \omega_1, \dots\}$
- ② Multiple Issues: $\Omega = \prod_{i=1}^{n_i} I_i$

Protocol Type

- ① Mediated
- ② Unmediated



Outline

1 Negotiation

2 Classic Results

3 References

Nash Bargaining Game: Solution

- Nash Point (1950): The point at which the product of surplus utility (above reservation value) of negotiators is maximized

$$\arg \max_{\omega_1, \omega_2} \prod_{i=1}^2 (\tilde{u}_i(\omega_i) - \tilde{u}_i(\phi))$$

Nash Bargaining Game: Solution

- Nash Point (1950): The point at which the product of surplus utility (above reservation value) of negotiators is maximized

$$\arg \max_{\omega_1, \omega_2} \prod_{i=1}^2 (\tilde{u}_i(\omega_i) - \tilde{u}_i(\phi))$$

- Kalai-Smorodinsky Point (1975): The Pareto outcome with equal ratios of achieved surplus utility and maximum feasible surplus utility

$$\arg \max_{\omega_1, \omega_2 \in F} (\omega_1 + \omega_2) \text{ s.t. } \left(\frac{\tilde{u}_1(\omega_1) - \tilde{u}_1(\phi)}{\tilde{u}_2(\omega_2) - \tilde{u}_2(\phi)} = \frac{\max_{v \in F} (\tilde{u}_1(v)) - \tilde{u}_1(\phi)}{\max_{v \in F} (\tilde{u}_2(v)) - \tilde{u}_2(\phi)} \right)$$

Nash Bargaining Game: Solution

- Nash Point (1950): The point at which the product of surplus utility (above reservation value) of negotiators is maximized

$$\arg \max_{\omega_1, \omega_2} \prod_{i=1}^2 (\tilde{u}_i(\omega_i) - \tilde{u}_i(\phi))$$

- Kalai-Smorodinsky Point (1975): The Pareto outcome with equal ratios of achieved surplus utility and maximum feasible surplus utility

$$\arg \max_{\omega_1, \omega_2 \in F} (\omega_1 + \omega_2) \text{ s.t. } \left(\frac{\tilde{u}_1(\omega_1) - \tilde{u}_1(\phi)}{\tilde{u}_2(\omega_2) - \tilde{u}_2(\phi)} = \frac{\max_{v \in F} (\tilde{u}_1(v)) - \tilde{u}_1(\phi)}{\max_{v \in F} (\tilde{u}_2(v)) - \tilde{u}_2(\phi)} \right)$$

- Kalai Point (1977): The Pareto outcome maximizing the utility for the unfortunate player. Defining P as the Pareto front

$$\arg \max_{\omega_1, \omega_2 \in P} \min_{i \in \{1, 2\}} (\tilde{u}_i(\omega_i) - \tilde{u}_i(\phi))$$

Rubinstein's Bargaining Protocol: Description

- Two agents sharing a pie.

Rubinstein's Bargaining Protocol: Description

- Two agents sharing a pie.
- Each agent is under a different time-pressure: $\tilde{u}_i^{t+\Delta}(\omega) < \tilde{u}_i^t(\omega)$.
Examples of time-pressure:

Rubinstein's Bargaining Protocol: Description

- Two agents sharing a pie.
- Each agent is under a different time-pressure: $\tilde{u}_i^{t+\Delta}(\omega) < \tilde{u}_i^t(\omega)$.

Examples of time-pressure:

Exponential $\tilde{u}_i^{t+\Delta}(\omega) = \delta_i^\Delta u_i^t(\omega)$.

Linear $\tilde{u}_i^{t+\Delta}(\omega) = u_i^t(\omega) - \Delta c_i$

Rubinstein's Bargaining Protocol: Description

- Two agents sharing a pie.
- Each agent is under a different time-pressure: $\tilde{u}_i^{t+\Delta}(\omega) < \tilde{u}_i^t(\omega)$.

Examples of time-pressure:

Exponential $\tilde{u}_i^{t+\Delta}(\omega) = \delta_i^\Delta u_i^t(\omega)$.

Linear $\tilde{u}_i^{t+\Delta}(\omega) = u_i^t(\omega) - \Delta c_i$

- Actor's initial utility is the assigned part of the pie: $\tilde{u}_i^0 = \omega_i$.

Rubinstein's Bargaining Protocol: Description

- Two agents sharing a pie.
- Each agent is under a different time-pressure: $\tilde{u}_i^{t+\Delta}(\omega) < \tilde{u}_i^t(\omega)$.

Examples of time-pressure:

Exponential $\tilde{u}_i^{t+\Delta}(\omega) = \delta_i^\Delta u_i^t(\omega)$.

Linear $\tilde{u}_i^{t+\Delta}(\omega) = u_i^t(\omega) - \Delta c_i$

- Actor's initial utility is the assigned part of the pie: $\tilde{u}_i^0 = \omega_i$.
- Time pressure and utility information are common knowledge.

Rubinstein's Bargaining Protocol: Description

- Two agents sharing a pie.
- Each agent is under a different time-pressure: $\tilde{u}_i^{t+\Delta}(\omega) < \tilde{u}_i^t(\omega)$.

Examples of time-pressure:

Exponential $\tilde{u}_i^{t+\Delta}(\omega) = \delta_i^\Delta u_i^t(\omega)$.

Linear $\tilde{u}_i^{t+\Delta}(\omega) = u_i^t(\omega) - \Delta c_i$

- Actor's initial utility is the assigned part of the pie: $\tilde{u}_i^0 = \omega_i$.
- Time pressure and utility information are common knowledge.
- No externally imposed time-limit.

Rubinstein's Bargaining Protocol: Description

- Two agents sharing a pie.
- Each agent is under a different time-pressure: $\tilde{u}_i^{t+\Delta}(\omega) < \tilde{u}_i^t(\omega)$.

Examples of time-pressure:

Exponential $\tilde{u}_i^{t+\Delta}(\omega) = \delta_i^\Delta u_i^t(\omega)$.

Linear $\tilde{u}_i^{t+\Delta}(\omega) = u_i^t(\omega) - \Delta c_i$

- Actor's initial utility is the assigned part of the pie: $\tilde{u}_i^0 = \omega_i$.
- Time pressure and utility information are common knowledge.
- No externally imposed time-limit.
- Zero reservation value: $u_i^\tau(\phi) = 0 \quad \forall \tau$.

Rubinstein's Bargaining Protocol: Description

- Two agents sharing a pie.
- Each agent is under a different time-pressure: $\tilde{u}_i^{t+\Delta}(\omega) < \tilde{u}_i^t(\omega)$.

Examples of time-pressure:

Exponential $\tilde{u}_i^{t+\Delta}(\omega) = \delta_i^\Delta u_i^t(\omega)$.

Linear $\tilde{u}_i^{t+\Delta}(\omega) = u_i^t(\omega) - \Delta c_i$

- Actor's initial utility is the assigned part of the pie: $\tilde{u}_i^0 = \omega_i$.
- Time pressure and utility information are common knowledge.
- No externally imposed time-limit.
- Zero reservation value: $u_i^\tau(\phi) = 0 \quad \forall \tau$.

Main Result

There is a unique *sub-game perfect equilibrium* that requires a single negotiation step in most cases.

Rubinstein's Bargaining Protocol: Equilibrium

Exponential Discounting

The negotiation ends in **one step** with the first agent proposing and the second agent accepting *for asymmetric cases*:

$$(\omega_1^*, \omega_2^*) = \left(\frac{1 - \delta_2}{1 - \delta_1 \delta_2}, \frac{\delta_2 (1 - \delta_1)}{1 - \delta_1 \delta_2} \right)$$

Rubinstein's Bargaining Protocol: Equilibrium

Exponential Discounting

The negotiation ends in **one step** with the first agent proposing and the second agent accepting *for asymmetric cases*:

$$(\omega_1^*, \omega_2^*) = \left(\frac{1 - \delta_2}{1 - \delta_1 \delta_2}, \frac{\delta_2 (1 - \delta_1)}{1 - \delta_1 \delta_2} \right)$$

Linear Discounting

The negotiation ends in **one step** with the first agent proposing and the second agent accepting:

$$(\omega_1^*, \omega_2^*) = \begin{cases} (c_2, 1 - c_2) & c_1 > c_2 \\ (x, 1 - x) \quad \forall x \in [c_1, 1] & c_1 = c_2 \\ (1, 0) & c_1 < c_2 \end{cases}$$

Negotiation With Full information

Hick's Paradox

Why do rational parties negotiate when they have full information?

Negotiation With Full information

Hick's Paradox

Why do rational parties negotiate when they have full information?

Because the world exists!! [Fernandez and Glazer, 1989]

Negotiation With Full information

Hick's Paradox

Why do rational parties negotiate when they have full information?

Because the world exists!! [Fernandez and Glazer, 1989]

- A union negotiating with management about a wage raise in rounds.
- The union *can* strike.
- Both parties are perfectly rational and fully informed.

Negotiation With Full information

Hick's Paradox

Why do rational parties negotiate when they have full information?

Because the world exists!! [Fernandez and Glazer, 1989]

- A union negotiating with management about a wage raise in rounds.
- The union *can* strike.
- Both parties are perfectly rational and fully informed.

Main Findings:

- Sub-game perfect equilibria exist in which there is some finite striking time followed by agreement.
- That happens in real time even when round length goes to zero.

Negotiation With Incomplete Information

Impossibility Result

Define a good mechanism as:

- Incentive compatible.
- No external subsidy.

Assuming rationality, there is *no* good mechanism that can guarantee agreement when it is dominant [Myerson and Satterthwaite, 1983].

Example

- A buyer values a product at v .
- A seller can create the product at cost c .
- $v > c$.
- There is no way to design a good mechanism that results in agreement for all v, c values.

Outline

1 Negotiation

2 Classic Results

3 References

References I

Fernandez, R. and Glazer, J. (1989). Striking for a bargain between two completely informed agents. Technical report, National Bureau of Economic Research.

Myerson, R. B. and Satterthwaite, M. A. (1983). Efficient mechanisms for bilateral trading. *Journal of economic theory*, 29(2):265–281.

Automated Negotiation: Challenges and Tools

Protocols and Strategies

Yasser Mohammad^{1,2,3} and Amy Greenwald⁴

¹ NEC CORPORATION, Global Innovation Unit

² National Institute of Advanced Industrial Science and Technology (AIST)

³ Assiut University, Egypt

⁴ Brown University

February 23rd, 2022

Outline

1 Negotiation Protocols

2 Strategies for SAOP

- Anatomy of a Negotiation Agent

3 Platforms Used in this Tutorial

4 NegMAS: The platform

5 References

Outline

1 Negotiation Protocols

2 Strategies for SAOP

3 Platforms Used in this Tutorial

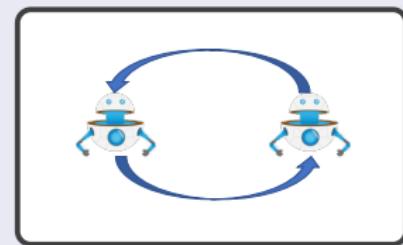
4 NegMAS: The platform

5 References

Unmediated Protocols

Main Features

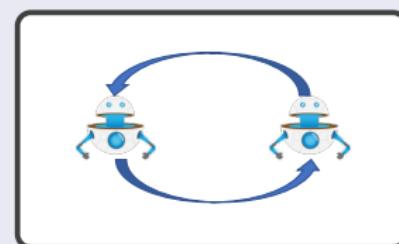
- No central coordinator.
- Agents negotiate by exchanging *messages*.
- All proposals come from negotiators.



Unmediated Protocols

Main Features

- No central coordinator.
- Agents negotiate by exchanging *messages*.
- All proposals come from negotiators.



Examples

Nash Bargaining Game Single iteration, single issue, bilateral protocol with complete information.

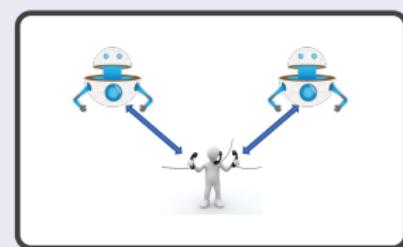
Rubinstein Bargaining Protocol Infinite horizon, single issue, bilateral protocol with complete information [Rubinstein, 1982].

Stacked Alternating Offers Protocol Finite horizon, multi-issue, multilateral protocol with partial information [Aydoğan et al., 2017].

Mediated Protocols

Main Features

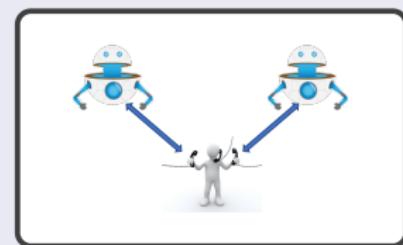
- Has A central *mediator*.
- Agents negotiate by exchanging messages with the *mediator*.
- Proposals can come from the mediator or the negotiators.



Mediated Protocols

Main Features

- Has A central *mediator*.
- Agents negotiate by exchanging messages with the *mediator*.
- Proposals can come from the mediator or the negotiators.



Examples

Single Text Protocol The mediator proposes a single hypothetical agreements, gets feedback about it and modifies it based on this feedback.

Stacked Alternating Offers Protocol

```
1     n_agreed, current = 0, randint(0, n_agents)
2     offer = agents[current].offer()
```

Stacked Alternating Offers Protocol

```
1      n_agreed, current = 0, randint(0, n_agents)
2      offer = agents[current].offer()
3
4      while True:
5          if timedout():
6              return 'TIME_OUT'
7          current = (current + 1) % n_agents
8          response = agents[current].respond(offer)
9          if response == 'accept':
10             n_agreed += 1
11             if n_agreed == n_agents:
12                 return offer # contract
13             elif response == 'end_negotiation':
14                 return 'FAILURE'
15             elif response == 'reject':
16                 offer = agents[current].offer()
```

Outline

1 Negotiation Protocols

2 Strategies for SAOP

- Anatomy of a Negotiation Agent

3 Platforms Used in this Tutorial

4 NegMAS: The platform

5 References

Outline

1 Negotiation Protocols

2 Strategies for SAOP

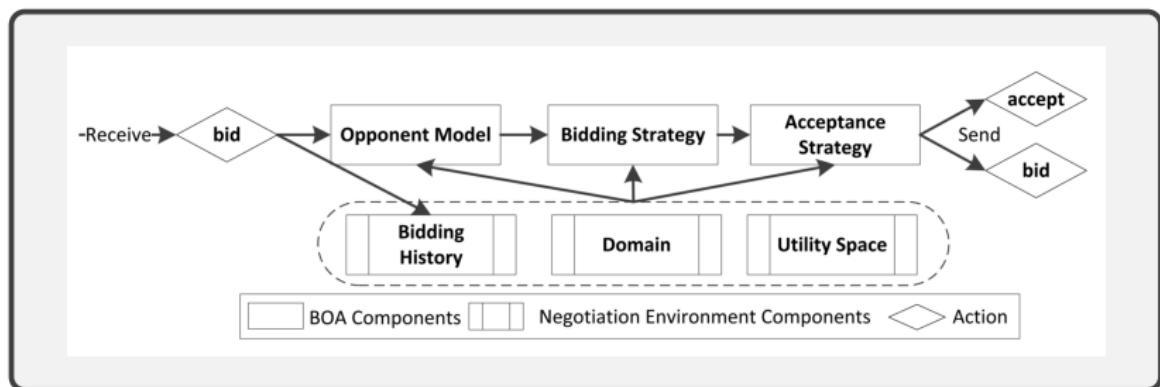
- Anatomy of a Negotiation Agent

3 Platforms Used in this Tutorial

4 NegMAS: The platform

5 References

Negotiator Components [Baarslag et al., 2014]¹



OBA Atchitecture

Opponent Model predicts opponent behavior.

Bidding Strategy Generates new bids.

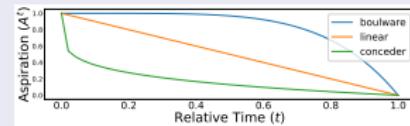
Acceptance Strategy Decides when to accept.

¹Supported by Genius

Bidding Strategy

Time-based strategies

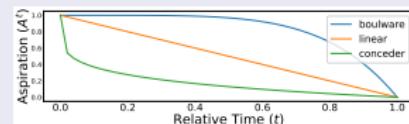
Offer an outcome with a utility just above the current *aspiration level* which is monotonically decreasing.



Bidding Strategy

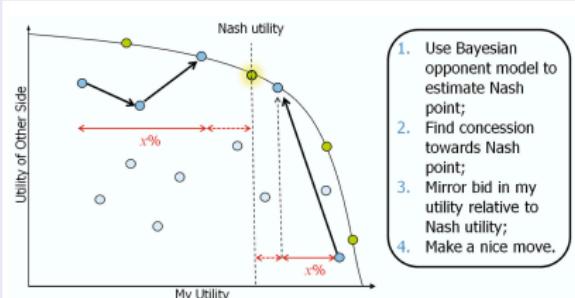
Time-based strategies

Offer an outcome with a utility just above the current *aspiration level* which is monotonically decreasing.



(Nice) Tit-for-Tat (bilateral) [Baarslag et al., 2013]

Concede as much as the opponent and do not retaliate.



Opponent Modeling

What is being modeled?

- Opponent preferences.
- Opponent strategy.
- Acceptance probability.
- Future offers.
- Opponent Type.

Opponent Modeling

What is being modeled?

- Opponent preferences.
- Opponent strategy.
- Acceptance probability.
- Future offers.
- Opponent Type.

When is it modeled?

- Before the negotiation.
- During the negotiation.

Opponent Modeling

What is being modeled?

- Opponent preferences.
- Opponent strategy.
- Acceptance probability.
- Future offers.
- Opponent Type.

When is it modeled?

- Before the negotiation.
- During the negotiation.

Data

- This negotiation vs. past negotiations.
- This opponent vs. this opponent group vs. others.
- Exchanged offers vs. agreements

Opponent Model: Example

Bayesian Learning

Hypothesis A hypothesis about the opponent's behavior.

Opponent Model: Example

Bayesian Learning

Hypothesis A hypothesis about the opponent's behavior.

Evidence Behavior of the agent (e.g. its offers/rejections).

Opponent Model: Example

Bayesian Learning

Hypothesis A hypothesis about the opponent's behavior.

Evidence Behavior of the agent (e.g. its offers/rejections).

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}$$

Opponent Model: Example

Bayesian Learning

Hypothesis A hypothesis about the opponent's behavior.

Evidence Behavior of the agent (e.g. its offers/rejections).

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}$$

Example

Hypothesis space: Utility function as a weighted sum of basis functions

$$u(\omega) = \sum_{i=1}^n \alpha_i f_i(\omega_i; \sigma_i)$$

Opponent Model: Example

Bayesian Learning

Hypothesis A hypothesis about the opponent's behavior.

Evidence Behavior of the agent (e.g. its offers/rejections).

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}$$

Example

Hypothesis space: Utility function as a weighted sum of basis functions

$$u(\omega) = \sum_{i=1}^n \alpha_i f_i(\omega_i; \sigma_i)$$

Evidence: Rejection and offers (assuming a strategy).

Acceptance Strategy

Examples

Accept if the utility of the offer \succ

Acceptance Strategy

Examples

Accept if the utility of the offer \succ

Previous my last offer.

Acceptance Strategy

Examples

Accept if the utility of the offer \succ

Previous my last offer.

Current what I am about to offer.

Acceptance Strategy

Examples

Accept if the utility of the offer \succ

Previous my last offer.

Current what I am about to offer.

Expected the best offer I expect to receive (needs an opponent model).

Acceptance Strategy

Examples

Accept if the utility of the offer \succ

Previous my last offer.

Current what I am about to offer.

Expected the best offer I expect to receive (needs an opponent model).

Threshold an offer with the current utility threshold (τ).

Acceptance Strategy

Examples

Accept if the utility of the offer \succ

Previous my last offer.

Current what I am about to offer.

Expected the best offer I expect to receive (needs an opponent model).

Threshold an offer with the current utility threshold (τ).

Constant May be a fraction of maximum utility.

Acceptance Strategy

Examples

Accept if the utility of the offer \succ

Previous my last offer.

Current what I am about to offer.

Expected the best offer I expect to receive (needs an opponent model).

Threshold an offer with the current utility threshold (τ).

Constant May be a fraction of maximum utility.

Time-based Monotonically decreasing with time.

Acceptance Strategy

Examples

Accept if the utility of the offer \succ

Previous my last offer.

Current what I am about to offer.

Expected the best offer I expect to receive (needs an opponent model).

Threshold an offer with the current utility threshold (τ).

Constant May be a fraction of maximum utility.

Time-based Monotonically decreasing with time.

Predictive Predicts the expected/max utility on rejection (e.g. Gaussian Process).

Outline

- 1 Negotiation Protocols
- 2 Strategies for SAOP
- 3 Platforms Used in this Tutorial
- 4 NegMAS: The platform
- 5 References

Platforms [Used in this tutorial]

Genius [Lin et al., 2014]

a Java-based negotiation platform to develop general negotiating agents and create negotiation scenarios. The platform can simulate negotiation sessions and tournaments and provides analytical tools to evaluate the agents' performance.

GENIUS

>> General Environment for Negotiation with Intelligent multi-purpose Usage Simulation.

NegMAS [Mohammad et al., 2019]

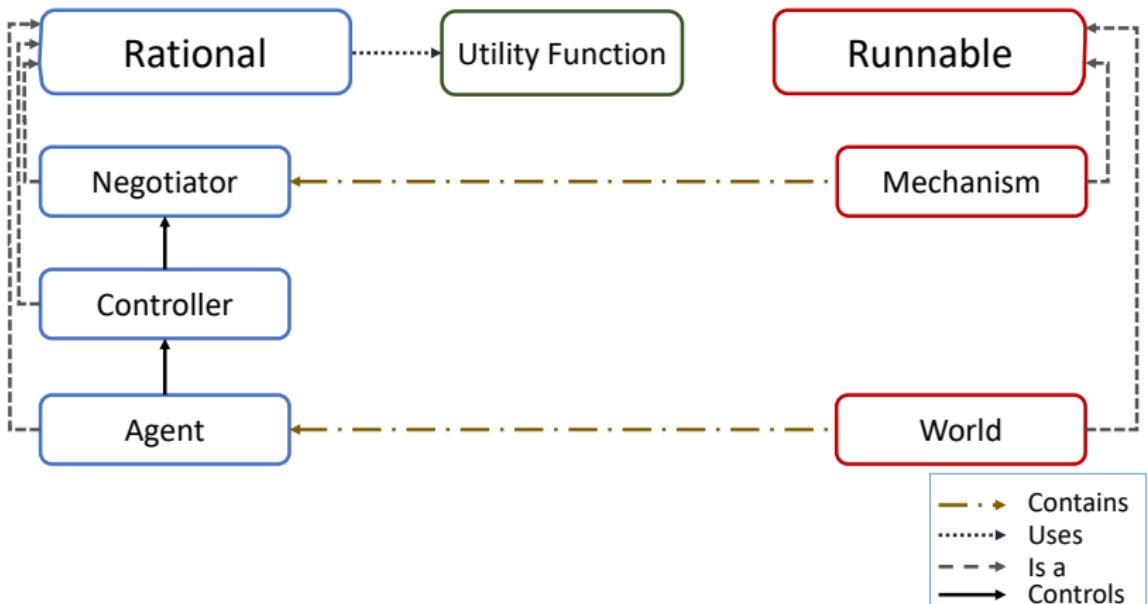
a Python-based negotiation platform for developing autonomous negotiation agents embedded in simulation environments. The main goal of NegMAS is to advance the state of the art in situated simultaneous negotiations.



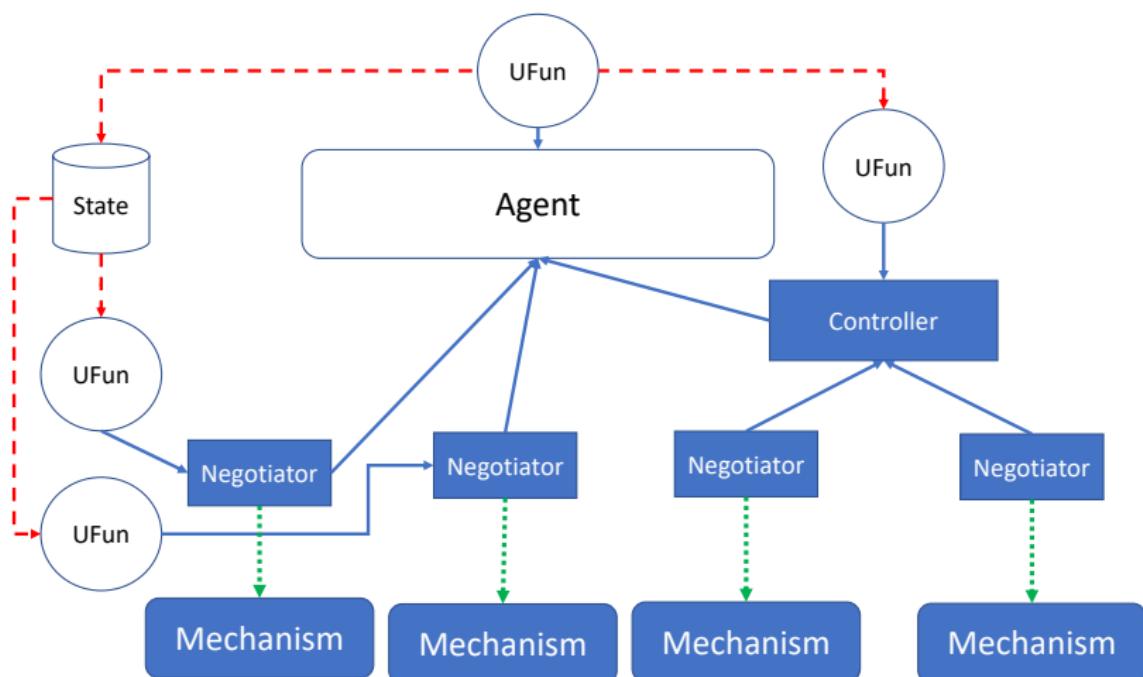
Outline

- 1 Negotiation Protocols
- 2 Strategies for SAOP
- 3 Platforms Used in this Tutorial
- 4 NegMAS: The platform
- 5 References

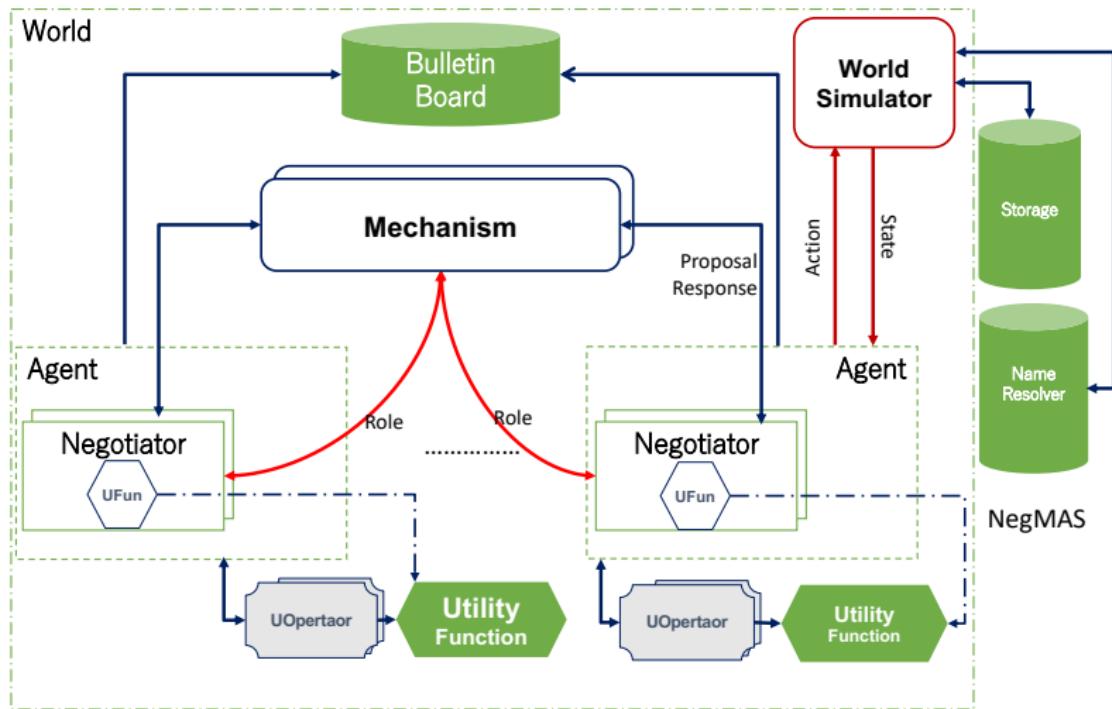
NegMAS² in two slides



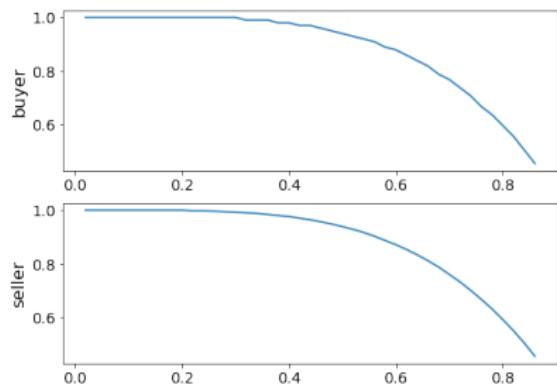
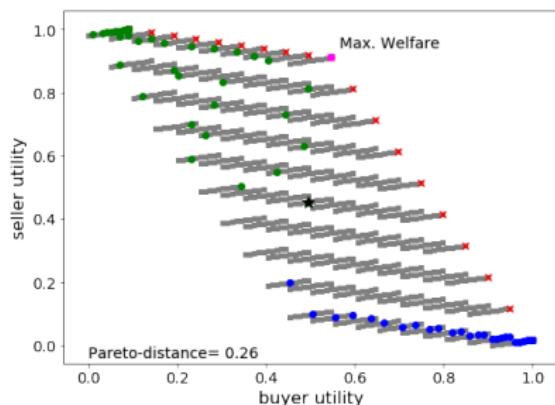
NegMAS in two slides



NegMAS in two slides (... OK 3)



NegMAS in two slides (... really!!!)



- An Example negotiation.
- Can you spot a problem?

Outline

- 1 Negotiation Protocols
- 2 Strategies for SAOP
- 3 Platforms Used in this Tutorial
- 4 NegMAS: The platform
- 5 References

References I

- Aydoğan, R., Festen, D., Hindriks, K. V., and Jonker, C. M. (2017). Alternating offers protocols for multilateral negotiation. In *Modern Approaches to Agent-based Complex Automated Negotiation*, pages 153–167. Springer.
- Baarslag, T., Hindriks, K., Hendrikx, M., Dirkzwager, A., and Jonker, C. (2014). *Decoupling Negotiating Agents to Explore the Space of Negotiation Strategies*, pages 61–83. Springer Japan, Tokyo.
- Baarslag, T., Hindriks, K., and Jonker, C. (2013). A tit for tat negotiation strategy for real-time bilateral negotiations. In *Complex Automated Negotiations: Theories, Models, and Software Competitions*, pages 229–233. Springer.
- Lin, R., Kraus, S., Baarslag, T., Tykhonov, D., Hindriks, K., and Jonker, C. M. (2014). Genius: An integrated environment for supporting the design of generic automated negotiators. *Computational Intelligence*, 30(1):48–70.
- Mohammad, Y., Greenwald, A., and Nakadai, S. (2019). Negmas: A platform for situated negotiations. In *Twelfth International Workshop on Agent-based Complex Automated Negotiations (ACAN2019) in conjunction with IJCAI*.
- Rubinstein, A. (1982). Perfect equilibrium in a bargaining model. *Econometrica: Journal of the Econometric Society*, pages 97–109.

Automated Negotiation: Challenges and Tools

Recent Advances

Yasser Mohammad^{1,2,3} and Amy Greenwald⁴

¹ NEC CORPORATION, Global Innovation Unit

² National Institute of Advanced Industrial Science and Technology (AIST)

³ Assiut University, Egypt

⁴ Brown University

February 23rd, 2022

Outline

- ① Negotiation with Incomplete Information
- ② Recent Optimality Results
- ③ Supervised Learning in Automated Negotiation
- ④ Reinforcement Learning in Automated Negotiation
- ⑤ References

Outline

- 1 Negotiation with Incomplete Information
- 2 Recent Optimality Results
- 3 Supervised Learning in Automated Negotiation
- 4 Reinforcement Learning in Automated Negotiation
- 5 References

Outline

1 Negotiation with Incomplete Information

2 Recent Optimality Results

3 Supervised Learning in Automated Negotiation

4 Reinforcement Learning in Automated Negotiation

5 References

Outline

- 1 Negotiation with Incomplete Information
- 2 Recent Optimality Results
- 3 Supervised Learning in Automated Negotiation
- 4 Reinforcement Learning in Automated Negotiation
- 5 References

Outline

- 1 Negotiation with Incomplete Information
- 2 Recent Optimality Results
- 3 Supervised Learning in Automated Negotiation
- 4 Reinforcement Learning in Automated Negotiation
- 5 References

Outline

- 1 Negotiation with Incomplete Information
- 2 Recent Optimality Results
- 3 Supervised Learning in Automated Negotiation
- 4 Reinforcement Learning in Automated Negotiation
- 5 References

References I

Automated Negotiation: Challenges and Tools

Automagted Negotiation is SCM

Yasser Mohammad^{1,2,3} and Amy Greenwald⁴

¹ NEC CORPORATION, Global Innovation Unit

² National Institute of Advanced Industrial Science and Technology (AIST)

³ Assiut University, Egypt

⁴ Brown University

February 23rd, 2022

Outline

1 Why Negotiation + SCM?

2 World Description

3 Agent

4 ANAC: A brief history

5 References

Outline

1 Why Negotiation + SCM?

2 World Description

3 Agent

4 ANAC: A brief history

5 References

Negotiation in SCM Business

CONTRACTROOM

pactum

- Human negotiations lead to an estimated 17-40% *value leakage* in some estimates¹

¹KPMG report: <https://bit.ly/3kDRy6I>

²Forrester report: <https://bit.ly/3nwXEaY>

³UN/CEFACT Project website: <https://bit.ly/38LOsLX>

Negotiation in SCM Business

CONTRACTROOM

pactum

- Human negotiations lead to an estimated 17-40% *value leakage* in some estimates ¹
- A recent study suggests that at least 15 companies are working in *contracting support systems* ².

¹KPMG report: <https://bit.ly/3kDRy6I>

²Forrester report: <https://bit.ly/3nwXEaY>

³UN/CEFACT Project website: <https://bit.ly/38LOsLX>

Negotiation in SCM Business

CONTRACT ROOMpactum

- Human negotiations lead to an estimated 17-40% *value leakage* in some estimates ¹
- A recent study suggests that at least 15 companies are working in *contracting support systems* ².
- A recent UNECE UN/CEFACT proposal to standardize negotiation protocols for SCM and other applications ³

¹KPMG report: <https://bit.ly/3kDRy6I>

²Forrester report: <https://bit.ly/3nwXEaY>

³UN/CEFACT Project website: <https://bit.ly/38LOsLX>

Negotiation in SCM Business

CONTRACTROOM**pactum**

- Human negotiations lead to an estimated 17-40% *value leakage* in some estimates ¹
- A recent study suggests that at least 15 companies are working in *contracting support systems* ².
- A recent UNECE UN/CEFACT proposal to standardize negotiation protocols for SCM and other applications ³
- More to come [Mohammad et al., 2019].

¹KPMG report: <https://bit.ly/3kDRy6I>

²Forrester report: <https://bit.ly/3nwXEaY>

³UN/CEFACT Project website: <https://bit.ly/38LOsLX>

Outline

1 Why Negotiation + SCM?

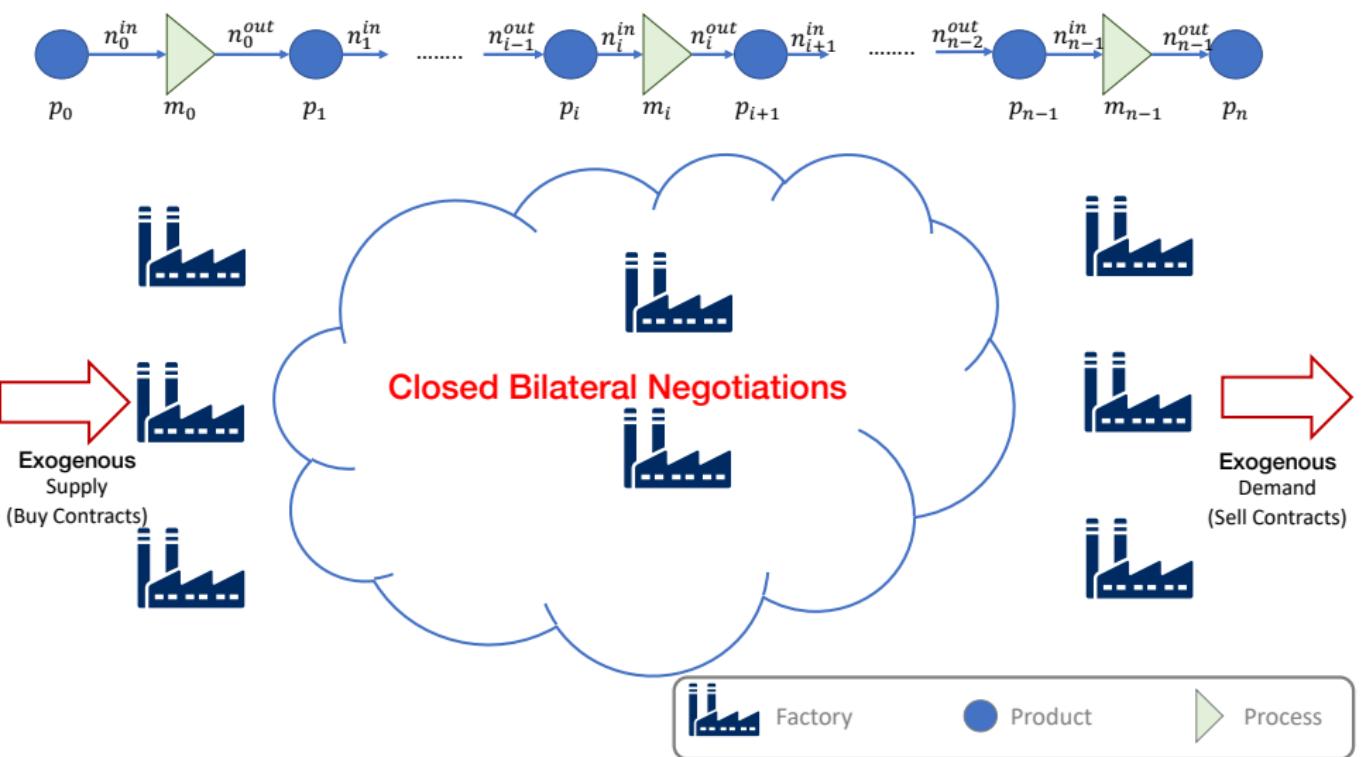
2 World Description

3 Agent

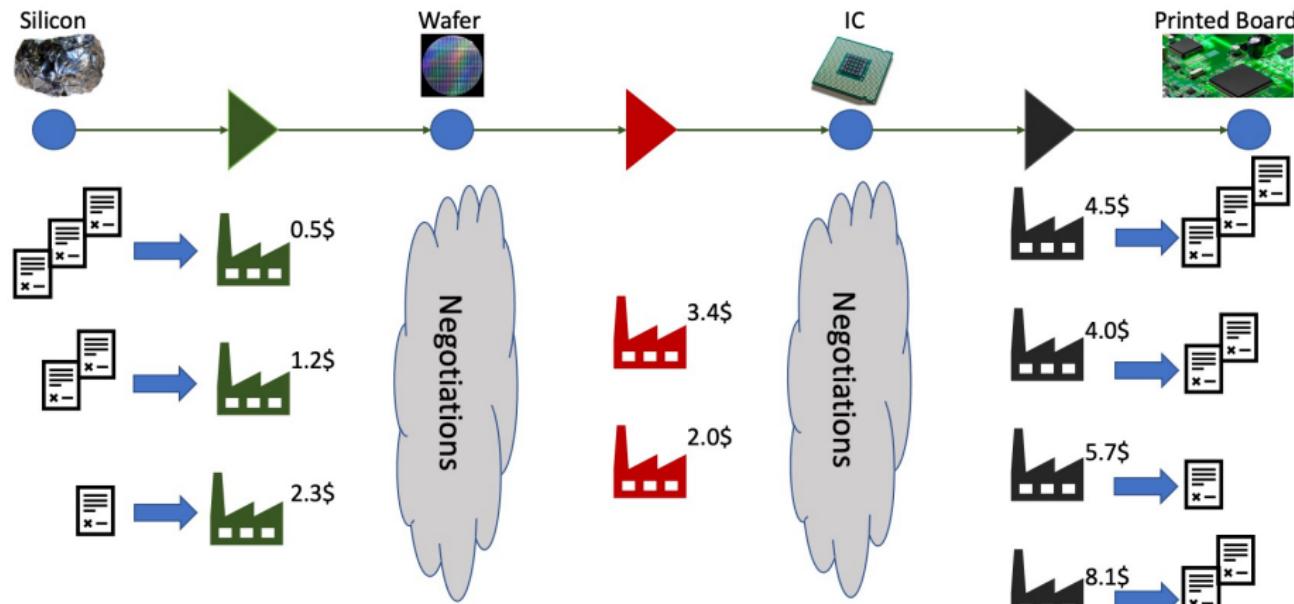
4 ANAC: A brief history

5 References

SCML Competition [Mohammad et al., 2019]



Example Configuration

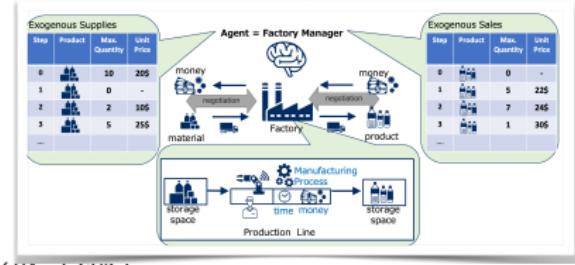
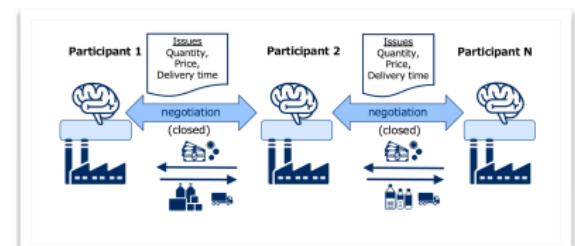
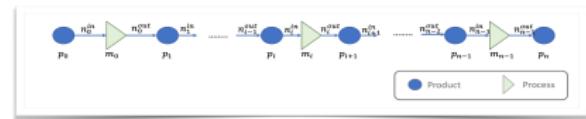


An example of an SCM world showing four products (circles), three processes (triangles) and few factories. Each process consumes one item of its input and generates one output of its output in one day. Each factory requires a different cost to run its process (shown in its top right). Factories in the first level have exogenous contracts to buy raw material (silicon) and factories at the last level have exogenous contracts to sell the final product (printed boards). These contracts drive the market.

SCML World

Challenge

- Turn maximize profit into a ufun!!
- Dynamic interdependent ufuncs.
- Sequential negotiations.
- Concurrent Negotiations.
- Negotiation under uncertainty.
- Adaptation and learning.
- Trust management.



Information

- Website** <https://scml.cs.brown.edu/>
- Code** <https://www.github.com/yasserfarouk/scml>
- Youtube** <https://www.youtube.com/playlist?list=PLqvs51K2Mb8IJe5Yz5jmYrRAwvIpGU2nF>

SCML Competition

Competition Details

- Runs as part of ANAC IJCAI.
- You control one or more factories.
 - Standard track** → one factory.
 - Collusion track** → multiple factories (3).

SCML Competition

Competition Details

- Runs as part of ANAC IJCAI.
- You control one or more factories.
 - Standard track** → one factory.
 - Collusion track** → multiple factories (3).

Score Evaluation

- Per instantiation:** Total profit counting inventory at **half** the **trading** price.
- Total:** **median** of per-instantiation scores.

SCML Competition

Competition Details

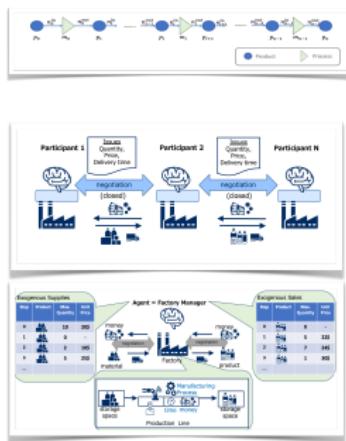
- Runs as part of ANAC IJCAI.
- You control one or more factories.
 - Standard track** → one factory.
 - Collusion track** → multiple factories (3).

Score Evaluation

- Per instantiation:** Total profit counting inventory at **half** the **trading** price.
- Total:** median of per-instantiation scores.

Flavors

- Online competition at
<https://scml.cs.brown.edu>
- Official competition as part of ANAC.



Leader Board

Rank	User	Score
1	user1	200
2	user2	100
3	user3	200
4	user4	100
5	user5	200

SCML 2020 League

One of the IJCAI 2020 Competition Logos

Information

Additional useful links, references, and other information about the competition.

ANAC 2020 was organized by a research institution in cooperation with the University of Texas at Dallas and the University of Illinois Urbana-Champaign. This document serves as a series of notes.

Important Dates

Important dates for the competition.

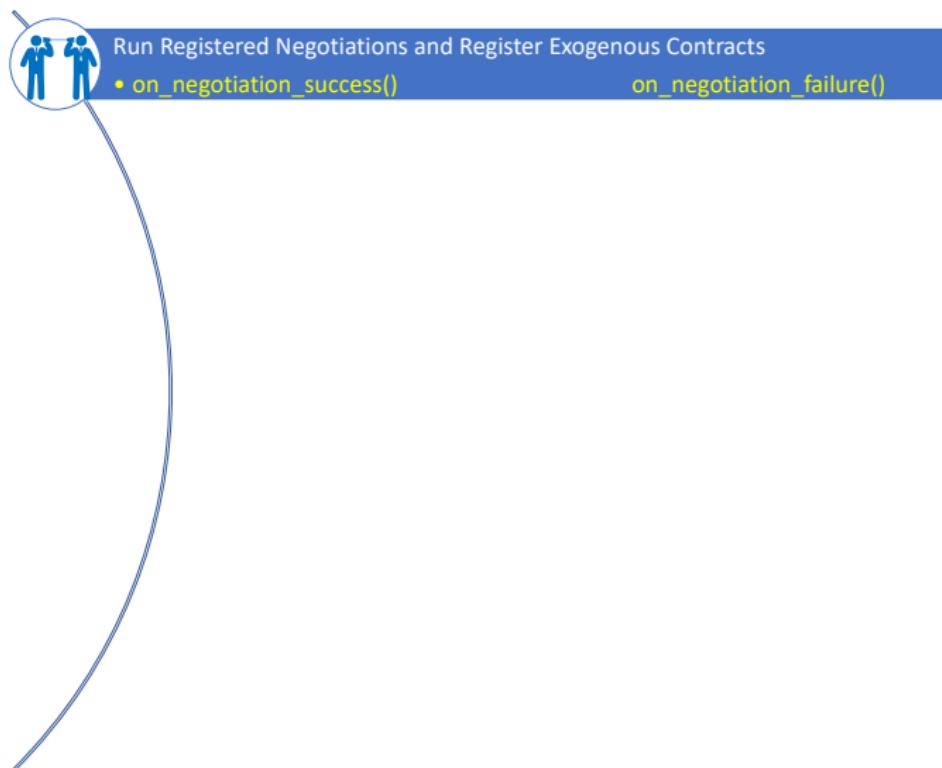
Simulation Steps

Simulation Steps



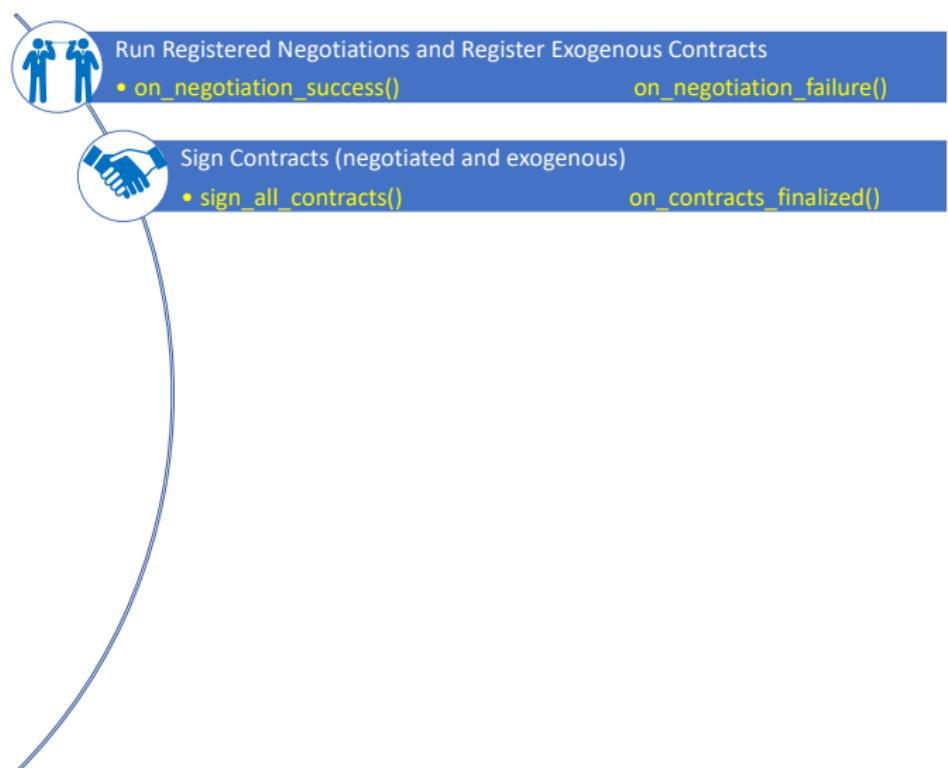
Simulation Steps

Simulation Steps



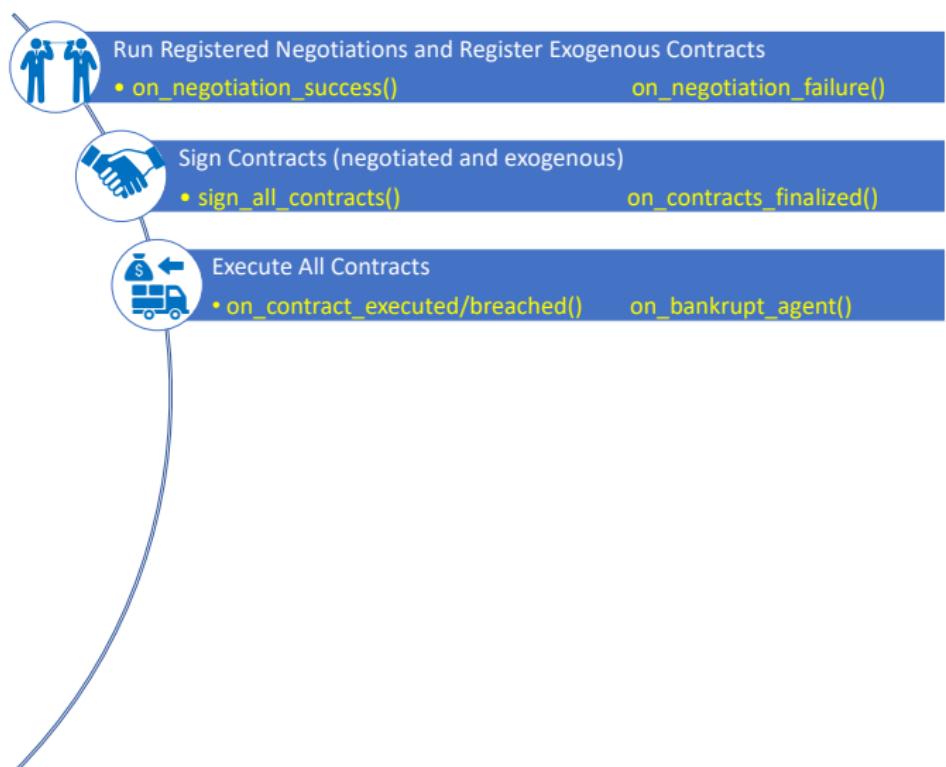
Simulation Steps

Simulation Steps



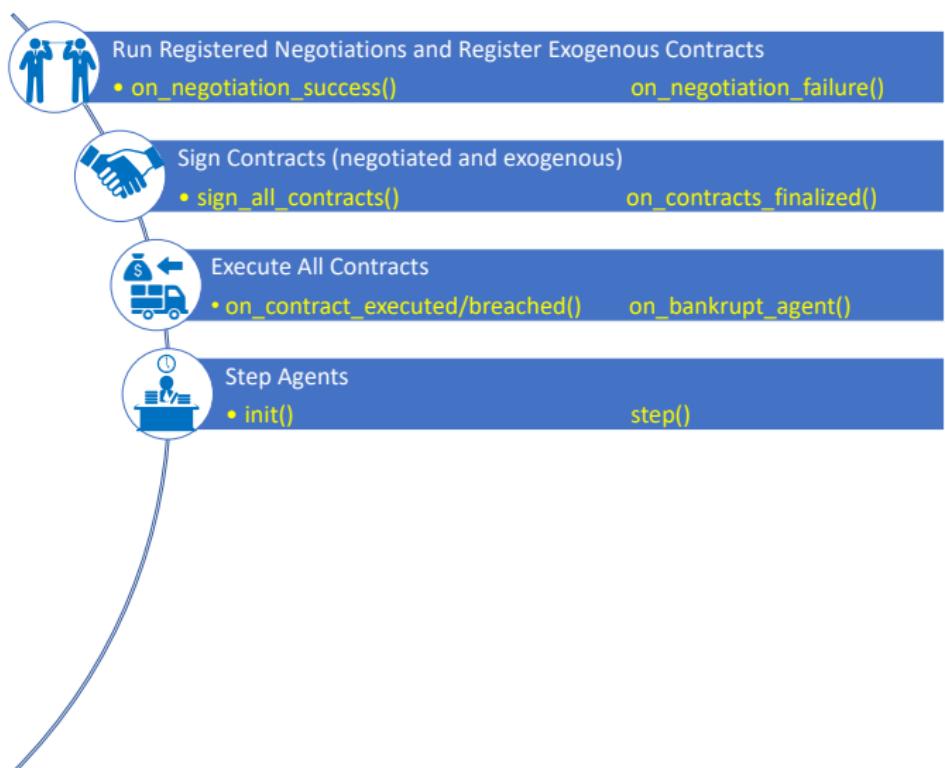
Simulation Steps

Simulation Steps



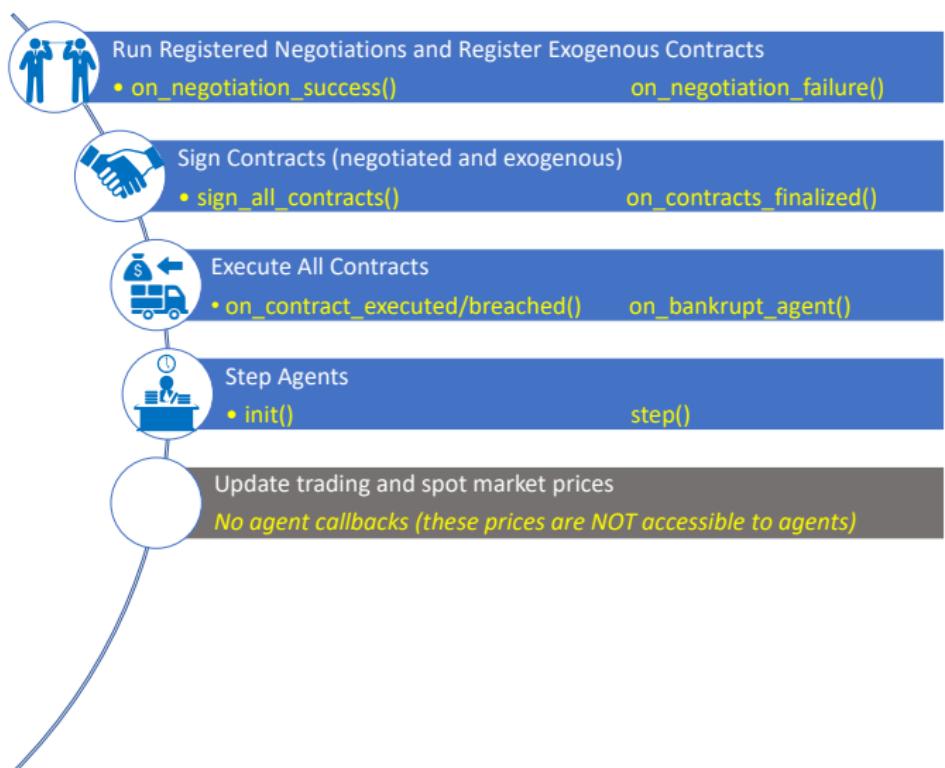
Simulation Steps

Simulation Steps



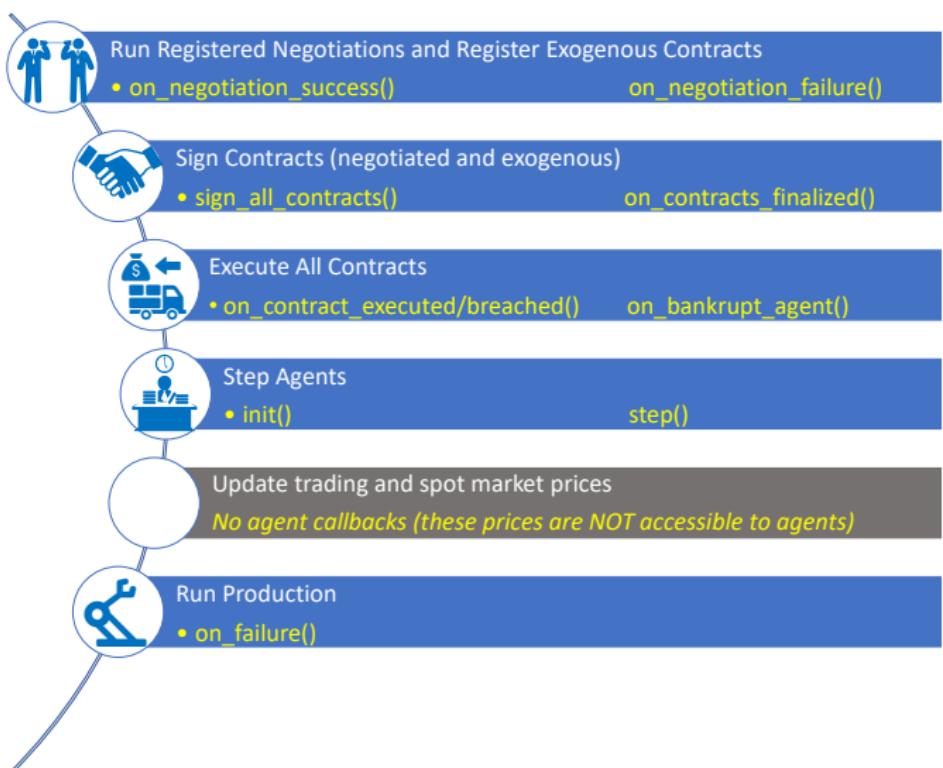
Simulation Steps

Simulation Steps



Simulation Steps

Simulation Steps

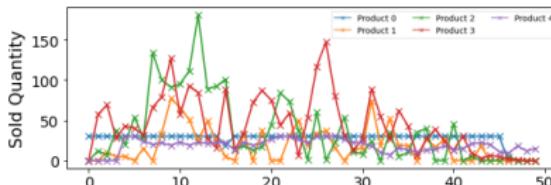
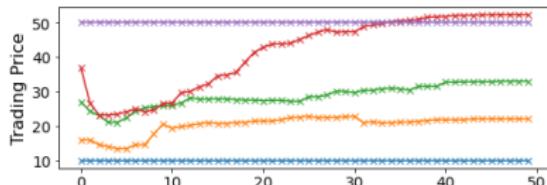


Simulation Steps

Simulation Steps



Trading prices



What is trading price and why is it calculated?

- A value calculated by the system **for each product**.
- Represents some estimate of the **current** price.
- **Never revealed** to agents.
- Usages:
 - Used at the end to value inventory.
 - Used when calculating **spot-market price**. during breach processing.

How does the system calculate it?

$$\text{tp}(p, s) = \frac{\beta^{s+1} Q_{-1}(p) \text{cat}(p) + \sum_{i=0}^s \beta^{s-i} Q_i(p) \mu_i(p)}{\beta^{s+1} + \sum_{i=0 | Q_i(p) > 0}^s \beta^{s-i}},$$

Trading prices: The details

Quantities and prices

$$Q_i(p') = \sum_{\{c \in C | c.p = p'\}} c.\bar{q}$$

$$\mu_i(p') = \frac{\sum_{\{c \in C | c.p = p'\}} c.\bar{q} \times c.u}{Q_i(p')}$$

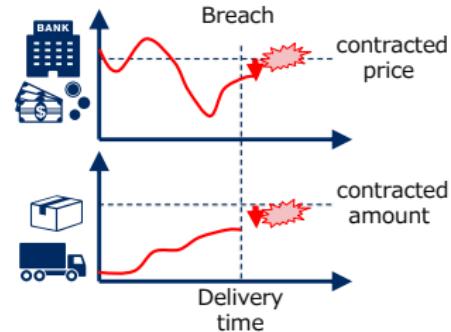
How does the system calculate it?

$$tp(p, s) = \frac{\beta^{s+1} Q_{-1}(p) \text{cat}(p) + \sum_{i=0}^s \beta^{s-i} Q_i(p) \mu_i(p)}{\beta^{s+1} + \sum_{i=0 | Q_i(p) > 0}^s \beta^{s-i}},$$

When things go wrong

What is a breach

- Insufficient funds or insufficient inventory.



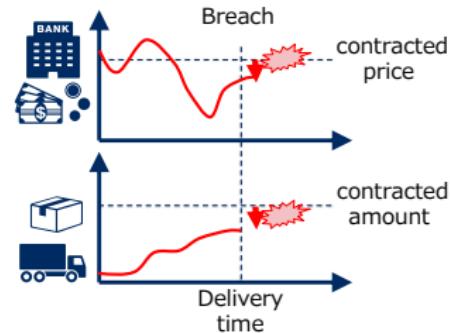
When things go wrong

What is a breach

- Insufficient funds or insufficient inventory.

Breach Processing

- A breach report is **always** published.
 - who and fraction.



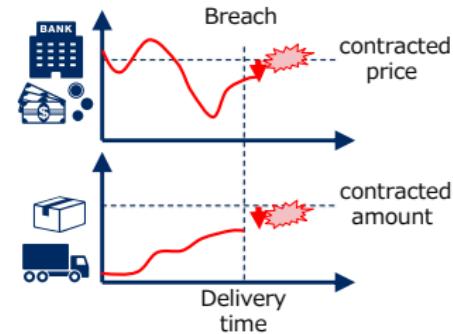
When things go wrong

What is a breach

- Insufficient funds or insufficient inventory.

Breach Processing

- A breach report is **always** published.
 - who and fraction.
- Insufficient products → **forced to buy** from the **spot market**.



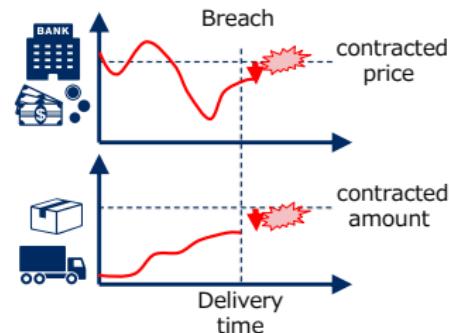
When things go wrong

What is a breach

- Insufficient funds or insufficient inventory.

Breach Processing

- A breach report is **always** published.
 - who and fraction.
- Insufficient products → **forced to buy** from the **spot market**.
- Insufficient funds → bankruptcy.



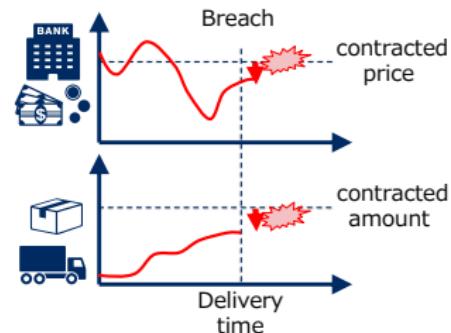
When things go wrong

What is a breach

- Insufficient funds or insufficient inventory.

Breach Processing

- A breach report is **always** published.
 - who and fraction.
- Insufficient products → **forced to buy** from the **spot market**.
- Insufficient funds → **bankruptcy**.
 - bankruptcy → liquidation.



Spot Market

Spot market

- **No-choice:** Can be used **only** for insufficient inventory breaches.

Calculation

Spot Market

Spot market

- **No-choice:** Can be used **only** for insufficient inventory breaches.
- **Penalizing:** Entails higher price than the **trading price**

Calculation

Spot Market

Spot market

- **No-choice:** Can be used **only** for insufficient inventory breaches.
- **Penalizing:** Entails higher price than the **trading price**
- **Personalized:** More breaches → higher spot price **for you**

Calculation

Spot Market

Spot market

- **No-choice:** Can be used **only** for insufficient inventory breaches.
- **Penalizing:** Entails higher price than the **trading price**
- **Personalized:** More breaches → higher spot price **for you**

Calculation

- Agent's spot price penalty: $\text{ip}_a(p, s) = \lambda \sum_{i=0}^s \alpha^{s-i} q_a(p, i)$,

Spot Market

Spot market

- **No-choice:** Can be used **only** for insufficient inventory breaches.
- **Penalizing:** Entails higher price than the **trading price**
- **Personalized:** More breaches → higher spot price **for you**

Calculation

- Agent's spot price penalty: $ip_a(p, s) = \lambda \sum_{i=0}^s \alpha^{s-i} q_a(p, i)$,
- Breaching agents buy **expensive**:

$$tp(p, s) \times (1 + gp) \times (1 + ip_a(p, s))$$

Spot Market

Spot market

- **No-choice:** Can be used **only** for insufficient inventory breaches.
- **Penalizing:** Entails higher price than the **trading price**
- **Personalized:** More breaches → higher spot price **for you**

Calculation

- Agent's spot price penalty: $ip_a(p, s) = \lambda \sum_{i=0}^s \alpha^{s-i} q_a(p, i)$,
- Breaching agents buy **expensive**:

$$tp(p, s) \times (1 + gp) \times (1 + ip_a(p, s))$$

- Bankrupt agents are liquidated **cheap**:

$$tp(p, s) / ((1 + gp) \times (1 + ip_a(p, s)))$$

Bankruptcy Processing

Bankruptcy conditions

- Insufficient money for a buy contract.

What exactly happens?

Bankruptcy Processing

Bankruptcy conditions

- Insufficient money for a buy contract.
- Insufficient money to buy from the spot market for a sell contract.

What exactly happens?

Bankruptcy Processing

Bankruptcy conditions

- Insufficient money for a buy contract.
- Insufficient money to buy from the spot market for a sell contract.

What exactly happens?

- ① The agent is stopped from every buying or selling.

Bankruptcy Processing

Bankruptcy conditions

- Insufficient money for a buy contract.
- Insufficient money to buy from the spot market for a sell contract.

What exactly happens?

- ① The agent is stopped from every buying or selling.
- ② Its inventory is sold on the spot market.

Bankruptcy Processing

Bankruptcy conditions

- Insufficient money for a buy contract.
- Insufficient money to buy from the spot market for a sell contract.

What exactly happens?

- ① The agent is stopped from every buying or selling.
- ② Its inventory is sold on the spot market.
- ③ All agents are informed.

Bankruptcy Processing

Bankruptcy conditions

- Insufficient money for a buy contract.
- Insufficient money to buy from the spot market for a sell contract.

What exactly happens?

- ① The agent is stopped from every buying or selling.
- ② Its inventory is sold on the spot market.
- ③ All agents are informed.
- ④ Agents with future contracts with it are informed about the expected level of breach.

Bankruptcy Processing

Bankruptcy conditions

- Insufficient money for a buy contract.
- Insufficient money to buy from the spot market for a sell contract.

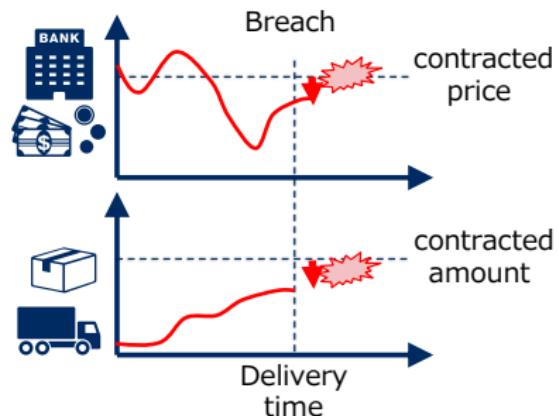
What exactly happens?

- ① The agent is stopped from every buying or selling.
- ② Its inventory is sold on the spot market.
- ③ All agents are informed.
- ④ Agents with future contracts with it are informed about the expected level of breach.
- ⑤ The agent's score is set to -1.

When things go wrong: Summary

Summary

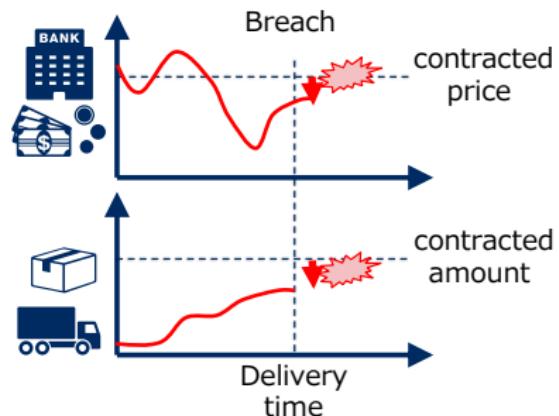
- An unfulfilled contract is reported to the **breach-list** (who and fraction).



When things go wrong: Summary

Summary

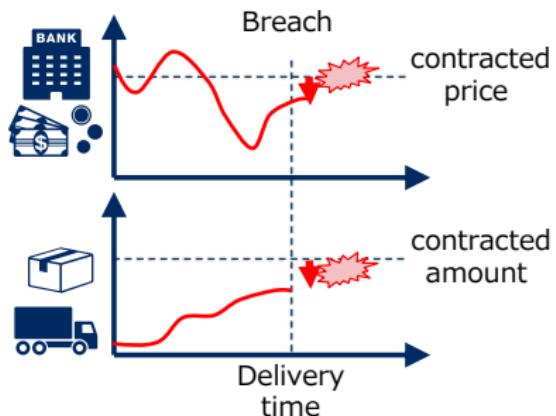
- An unfulfilled contract is reported to the **breach-list** (who and fraction).
- Insufficient funds → bankrupt.



When things go wrong: Summary

Summary

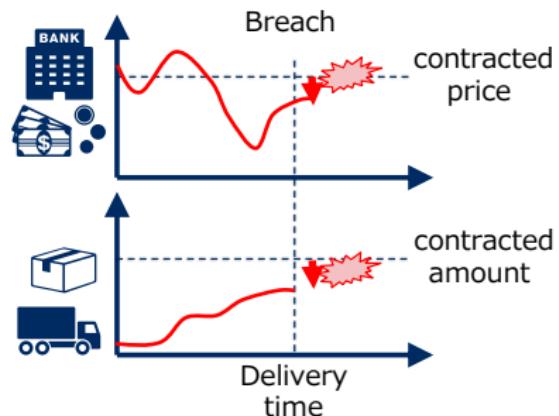
- An unfulfilled contract is reported to the **breach-list** (who and fraction).
- Insufficient funds → bankrupt.
- Insufficient product → buy at high cost.



When things go wrong: Summary

Summary

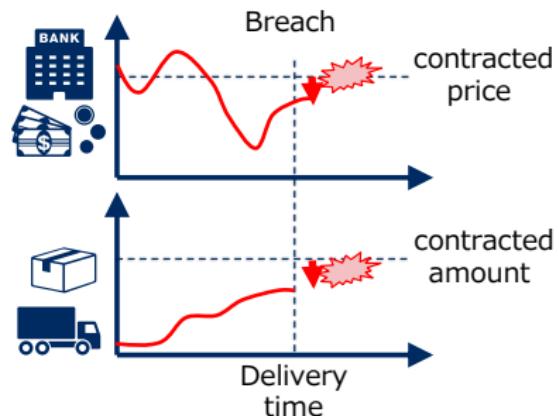
- An unfulfilled contract is reported to the **breach-list** (who and fraction).
- Insufficient funds → bankrupt.
- Insufficient product → buy at high cost.
 - Cannot buy → bankrupt.



When things go wrong: Summary

Summary

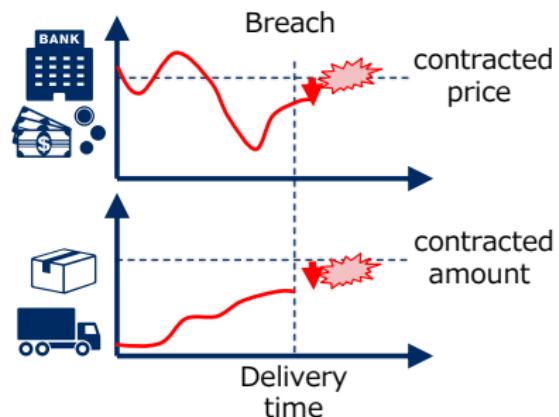
- An unfulfilled contract is reported to the **breach-list** (who and fraction).
- Insufficient funds → bankrupt.
- Insufficient product → buy at high cost.
 - Cannot buy → bankrupt.
- Bankrupt → **really really bad**



When things go wrong: Summary

Summary

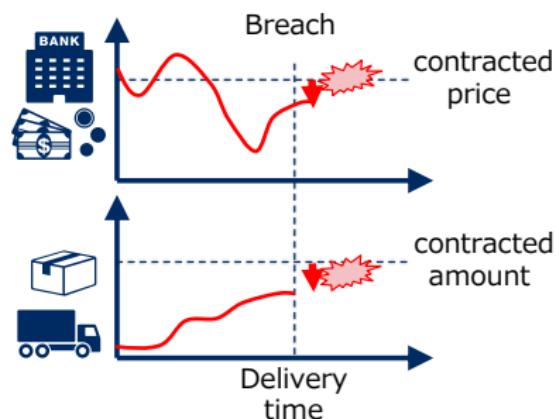
- An unfulfilled contract is reported to the **breach-list** (who and fraction).
- Insufficient funds → bankrupt.
- Insufficient product → buy at high cost.
 - Cannot buy → bankrupt.
- Bankrupt → **really really bad**
 - No more trade.



When things go wrong: Summary

Summary

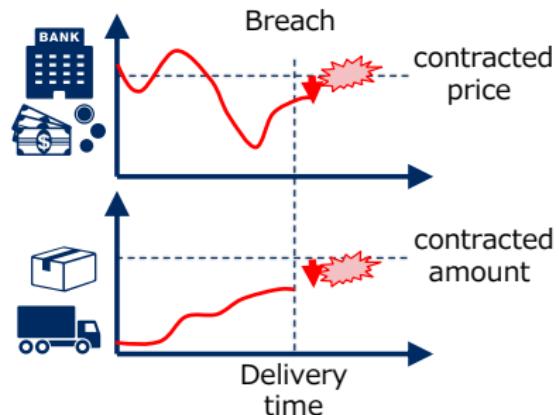
- An unfulfilled contract is reported to the **breach-list** (who and fraction).
- Insufficient funds → bankrupt.
- Insufficient product → buy at high cost.
 - Cannot buy → bankrupt.
- Bankrupt → **really really bad**
 - No more trade.
 - Very low score.



When things go wrong: Summary

Summary

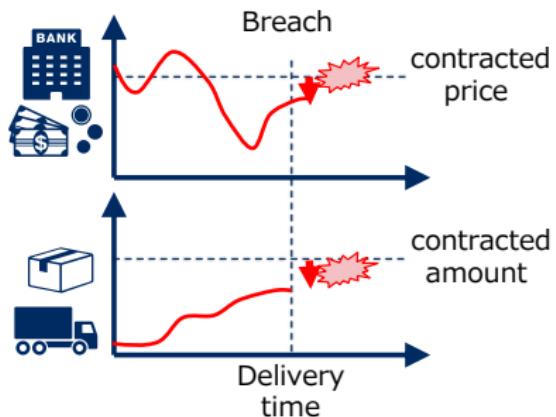
- An unfulfilled contract is reported to the **breach-list** (who and fraction).
- Insufficient funds → bankrupt.
- Insufficient product → buy at high cost.
 - Cannot buy → bankrupt.
- Bankrupt → **really really bad**
 - No more trade.
 - Very low score.
 - All inventory is liquidated.



When things go wrong: Summary

Summary

- An unfulfilled contract is reported to the **breach-list** (who and fraction).
- Insufficient funds → bankrupt.
- Insufficient product → buy at high cost.
 - Cannot buy → bankrupt.
- Bankrupt → **really really bad**
 - No more trade.
 - Very low score.
 - All inventory is liquidated.
 - May hurt other agents.



Outline

1 Why Negotiation + SCM?

2 World Description

3 Agent

4 ANAC: A brief history

5 References

Agent Knowledge

About itself

- **Its capabilities:** lines and production cost.
- **Its location:** input and output products.
- **Its partners:** suppliers and consumers (and competitors).
- **Its state:** inventory, wallet, contracts, and negotiations.

Agent Knowledge

About itself

- **Its capabilities:** lines and production cost.
- **Its location:** input and output products.
- **Its partners:** suppliers and consumers (and competitors).
- **Its state:** inventory, wallet, contracts, and negotiations.

About the market

- The production graph and factories at each level.
- Time and simulation length.

Agent Knowledge

About itself

- **Its capabilities:** lines and production cost.
- **Its location:** input and output products.
- **Its partners:** suppliers and consumers (and competitors).
- **Its state:** inventory, wallet, contracts, and negotiations.

About the market

- The production graph and factories at each level.
- Time and simulation length.

About others

- **Financial Reports:** balance, assets, breach fraction/probability.
- **Past Interactions:** negotiations and contracts between itself and that agent.

Information about previous simulations

- Agents **cannot** pass information between simulations.

Information about previous simulations

- Agents **cannot** pass information between simulations.
- We release datasets from the online competition including:

Information about previous simulations

- Agents **cannot** pass information between simulations.
- We release datasets from the online competition including:
 - World parameters.

Information about previous simulations

- Agents **cannot** pass information between simulations.
- We release datasets from the online competition including:
 - World parameters.
 - Agent types.

Information about previous simulations

- Agents **cannot** pass information between simulations.
- We release datasets from the online competition including:
 - World parameters.
 - Agent types.
 - Financial reports.

Information about previous simulations

- Agents **cannot** pass information between simulations.
- We release datasets from the online competition including:
 - World parameters.
 - Agent types.
 - Financial reports.
 - Contracts.

Information about previous simulations

- Agents **cannot** pass information between simulations.
- We release datasets from the online competition including:
 - World parameters.
 - Agent types.
 - Financial reports.
 - Contracts.
 - Breaches.

Information about previous simulations

- Agents **cannot** pass information between simulations.
- We release datasets from the online competition including:
 - World parameters.
 - Agent types.
 - Financial reports.
 - Contracts.
 - Breaches.
- This data is anonymous:

Information about previous simulations

- Agents **cannot** pass information between simulations.
- We release datasets from the online competition including:
 - World parameters.
 - Agent types.
 - Financial reports.
 - Contracts.
 - Breaches.
- This data is anonymous:
 - You **cannot** associate an agent type with a specific participant.

Information about previous simulations

- Agents **cannot** pass information between simulations.
- We release datasets from the online competition including:
 - World parameters.
 - Agent types.
 - Financial reports.
 - Contracts.
 - Breaches.
- This data is anonymous:
 - You **cannot** associate an agent type with a specific participant.
 - You **can** associate agent types across simulations.

Development Approaches

Monolithic Agent

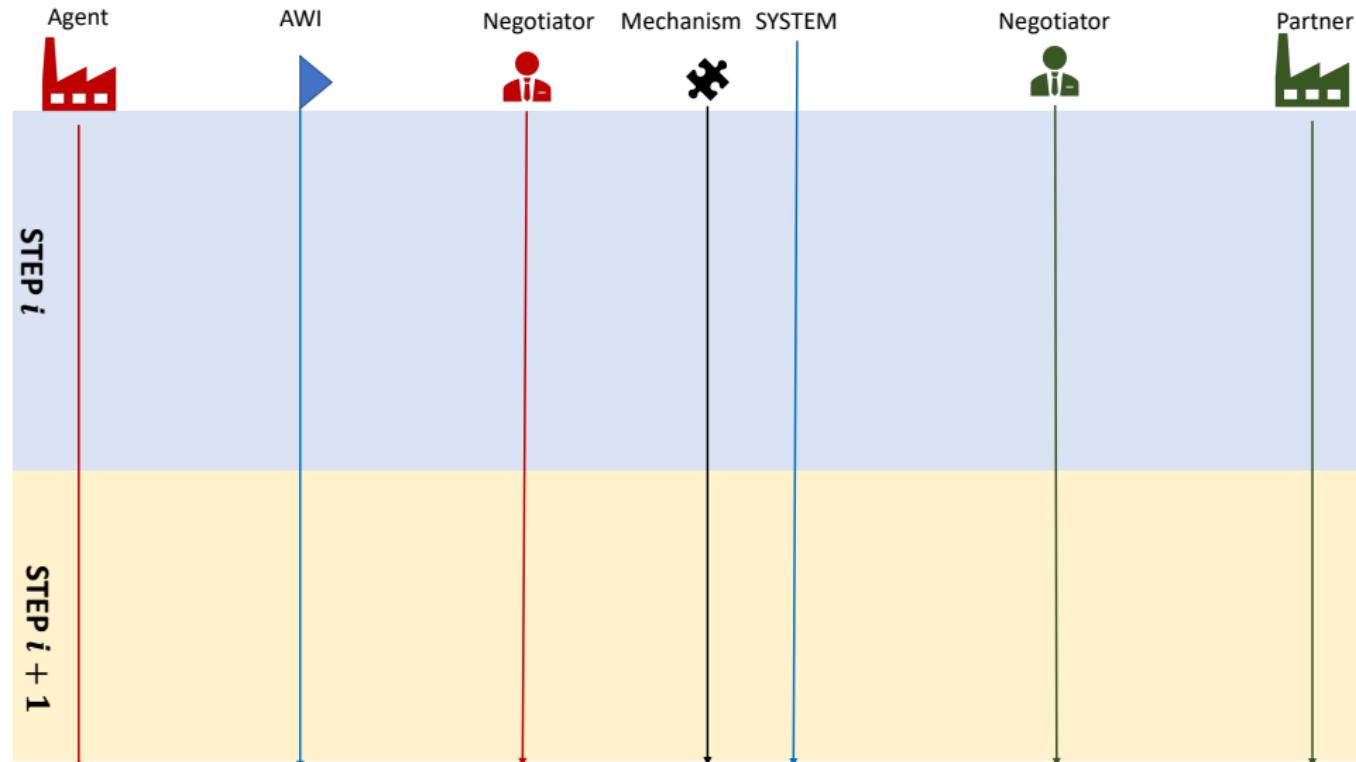
- Respond to callbacks in the **Agent** class.
- Functionality is distributed **among callbacks**.
- Everything is in one place (the agent class).
- Harder to reuse.

Component-Based Agent

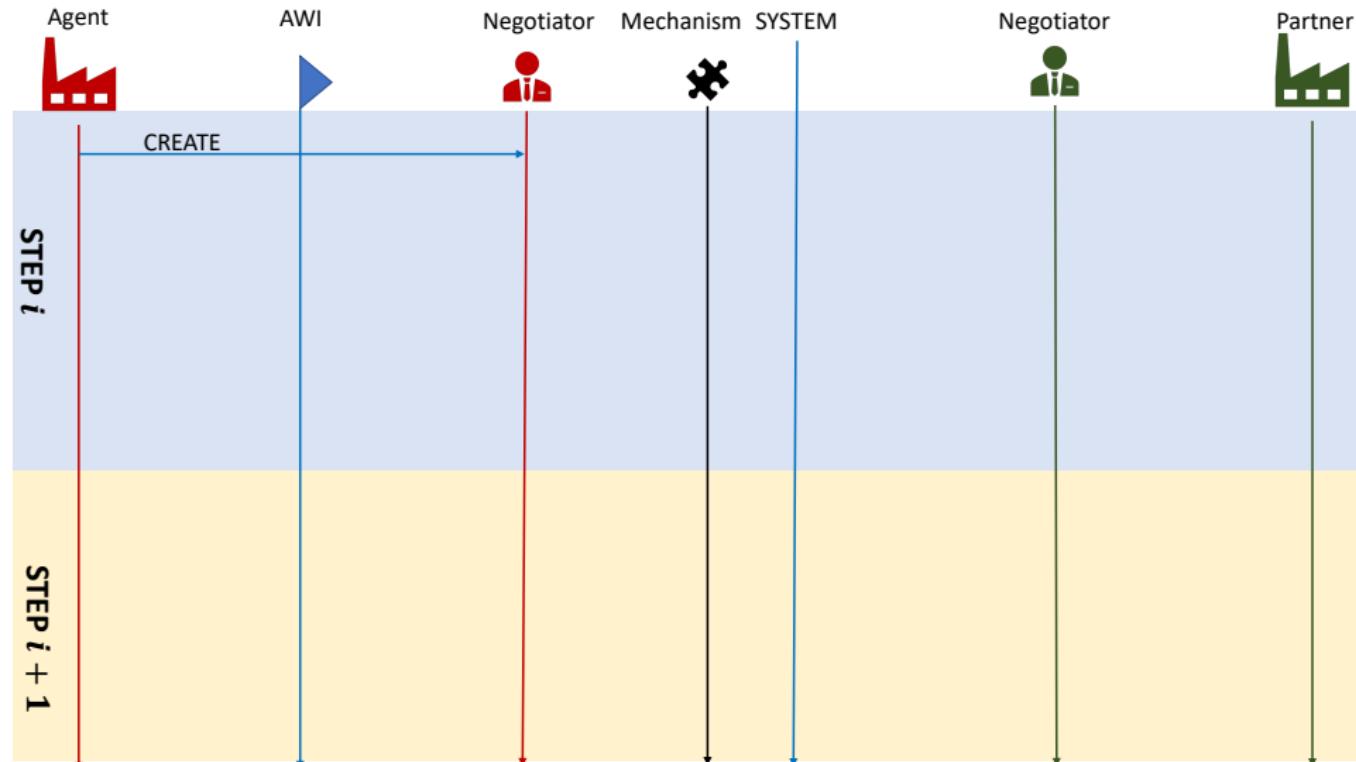
- Divides the agent into **semi-independent** components.
- Functionality is distributed between **components**.
- Easier to reuse.



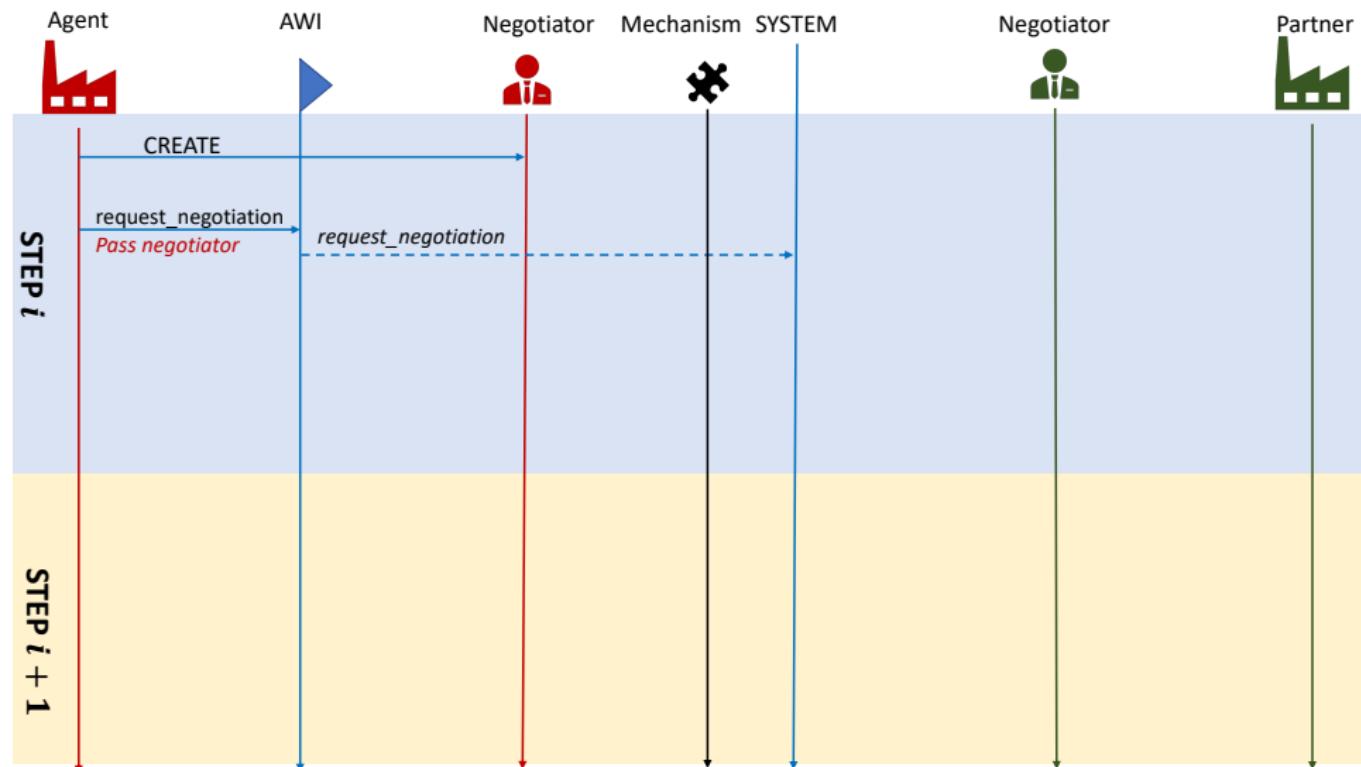
Callbacks and Timing



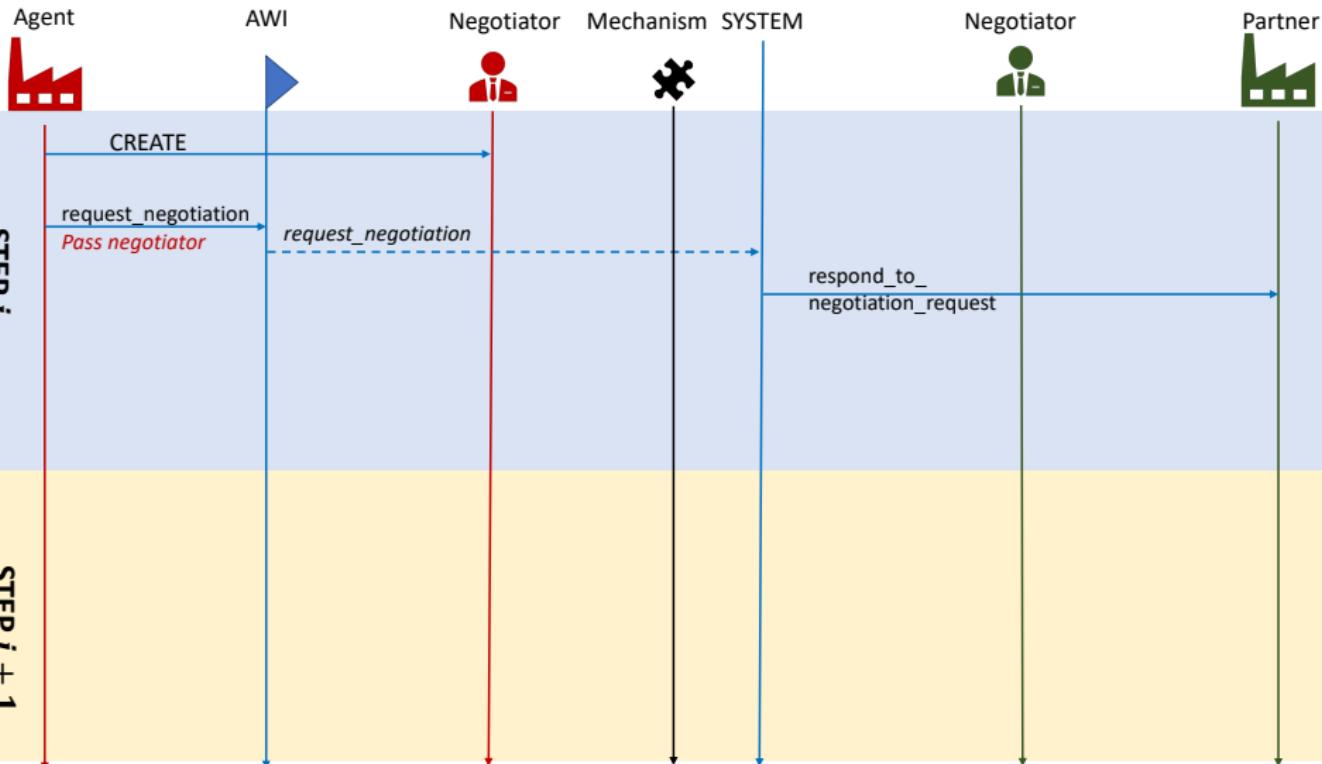
Callbacks and Timing



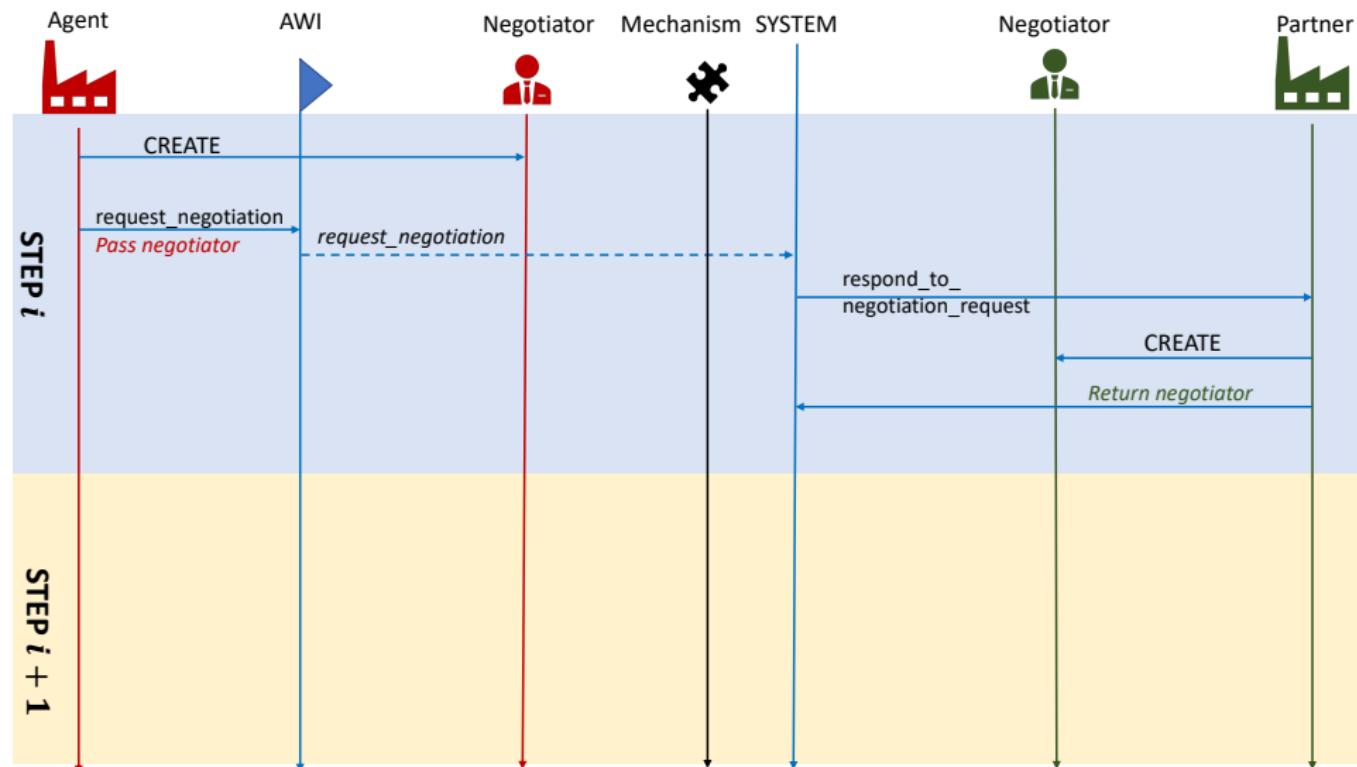
Callbacks and Timing



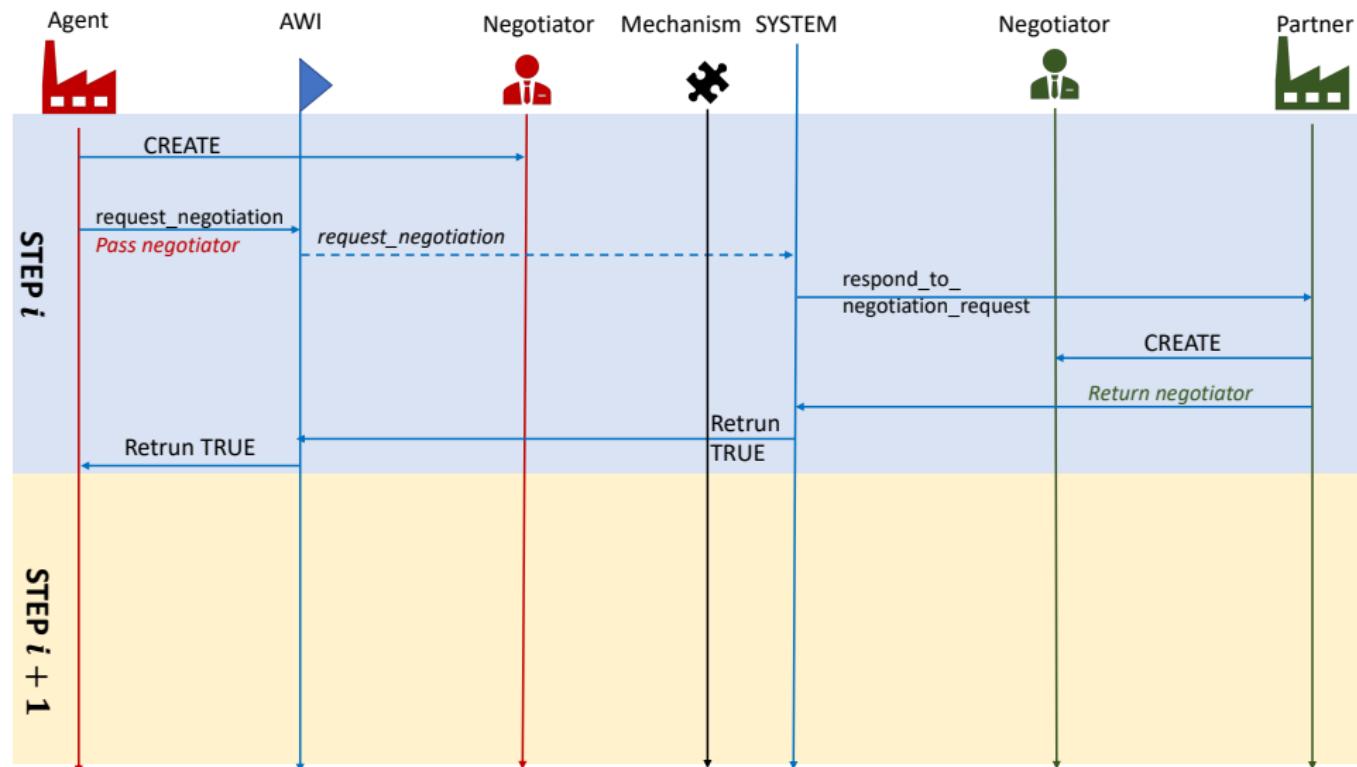
Callbacks and Timing



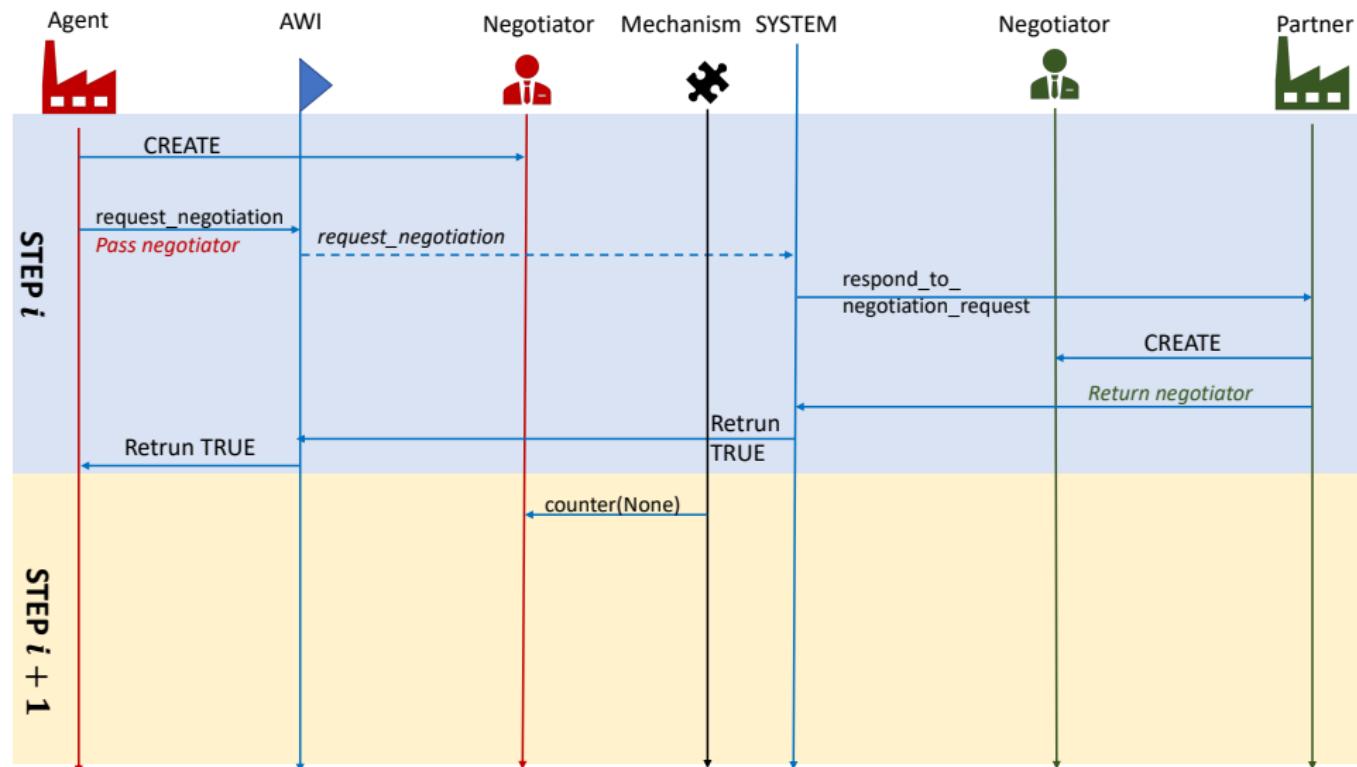
Callbacks and Timing



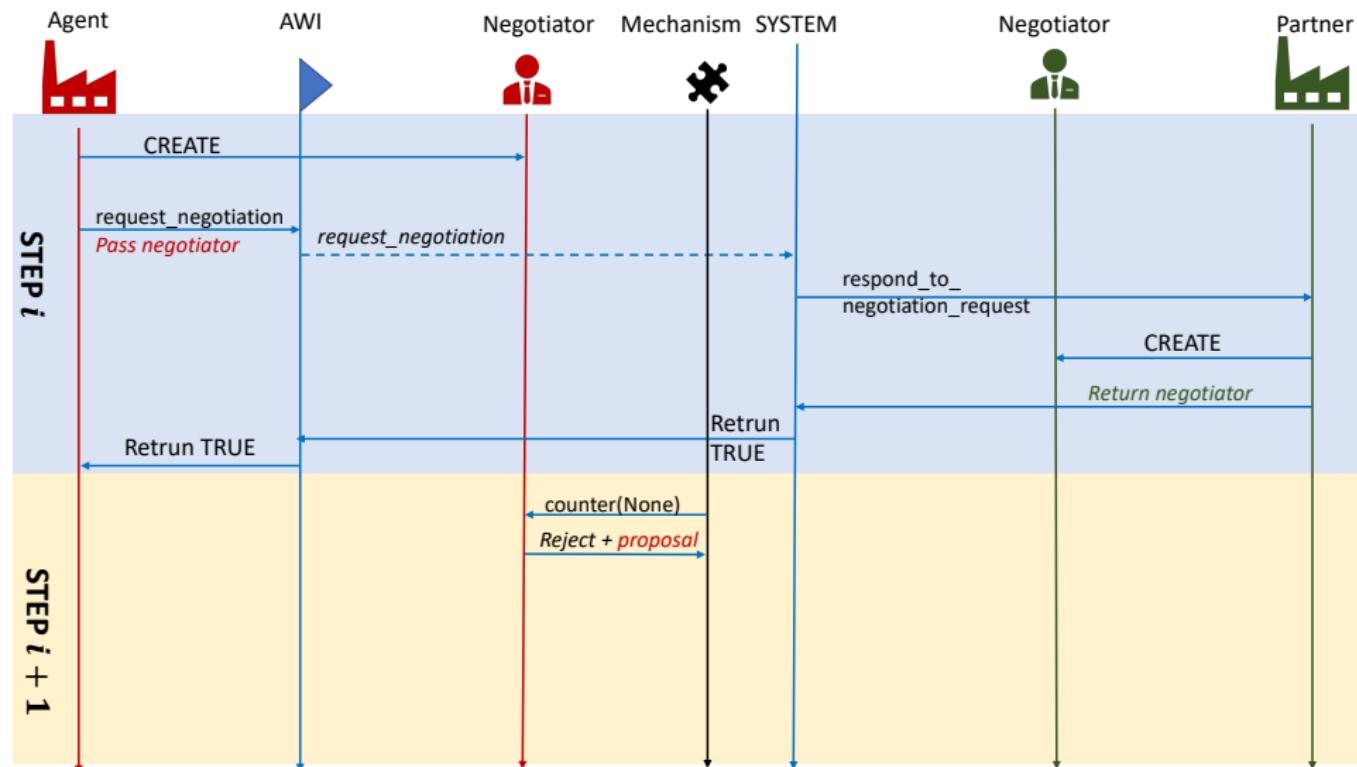
Callbacks and Timing



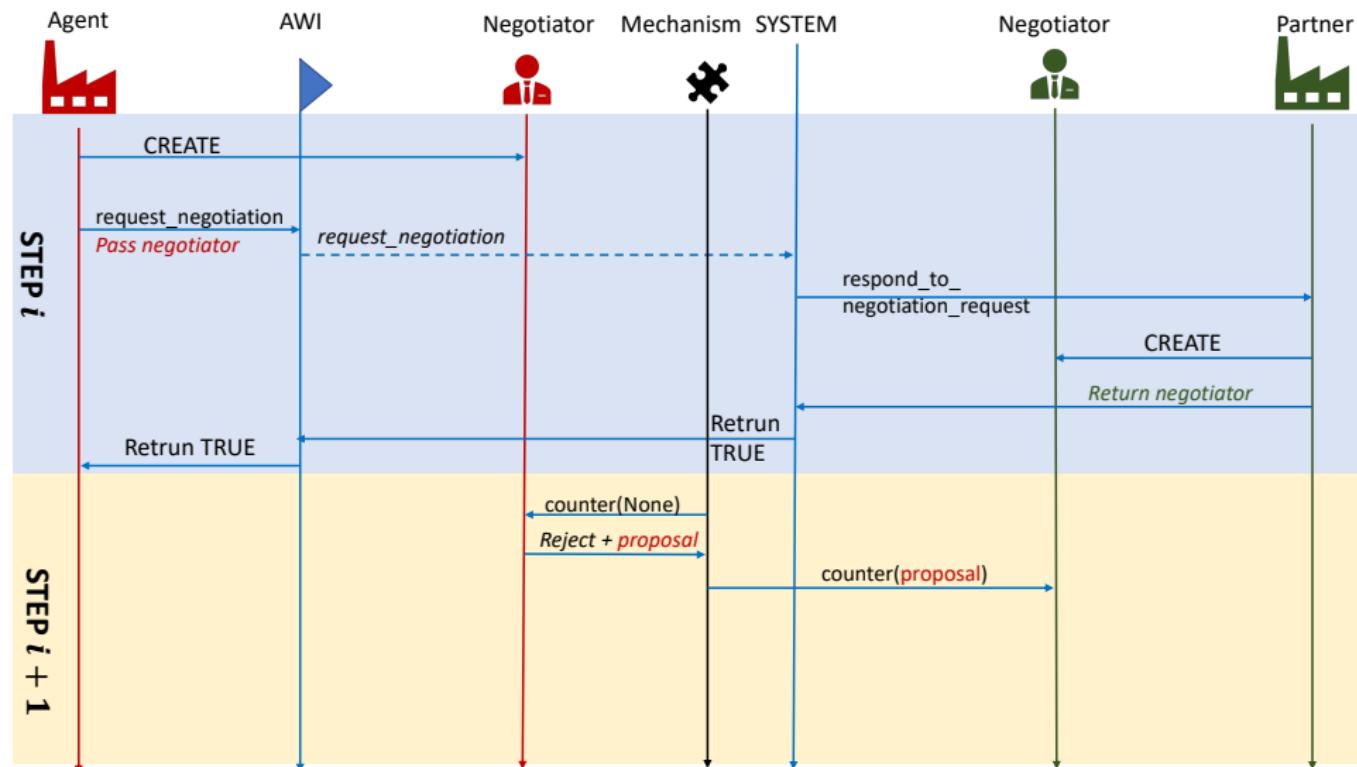
Callbacks and Timing



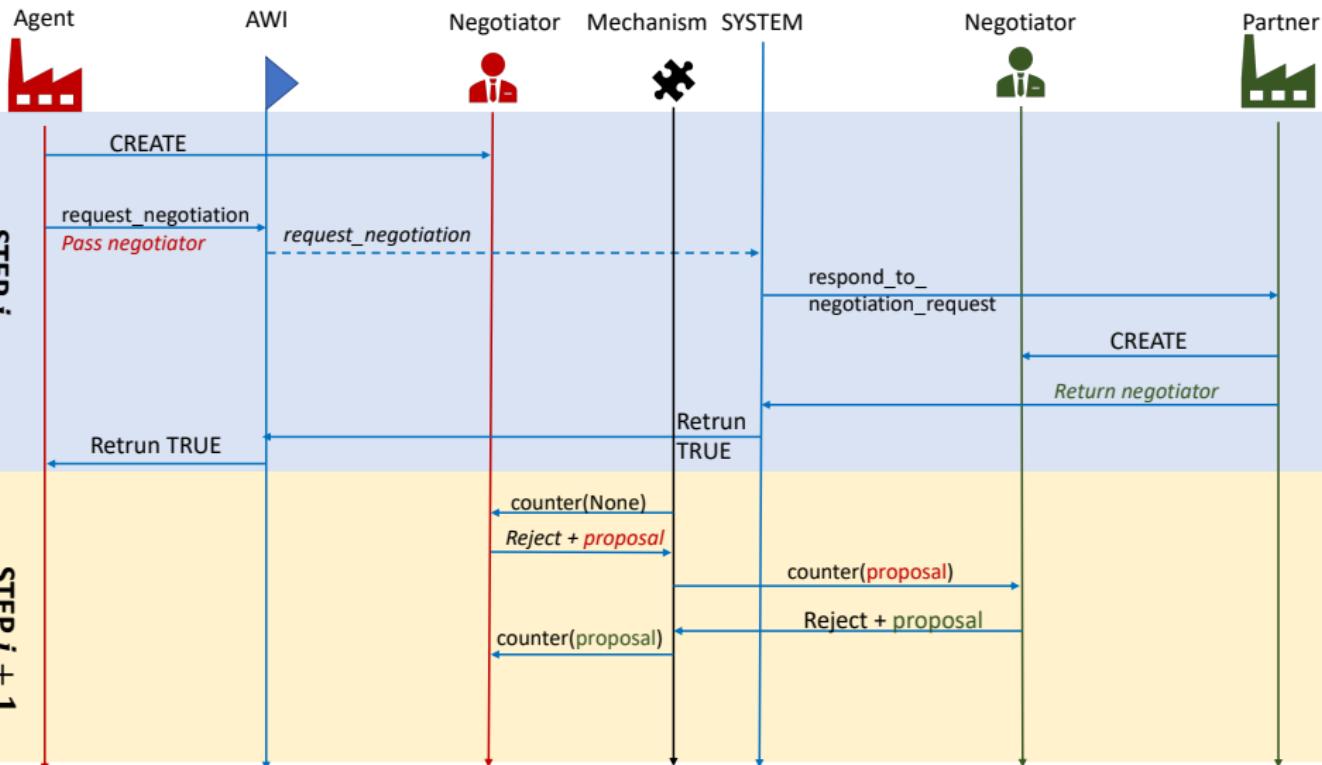
Callbacks and Timing



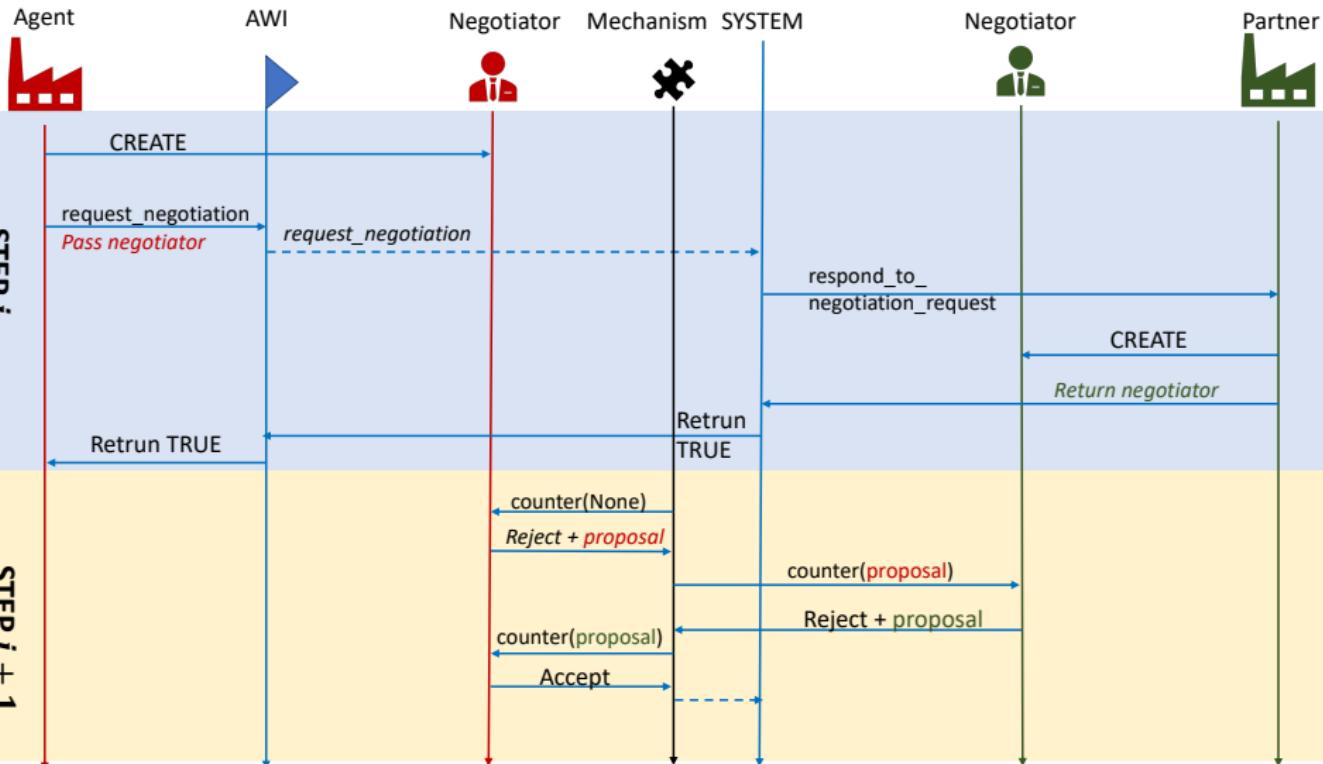
Callbacks and Timing



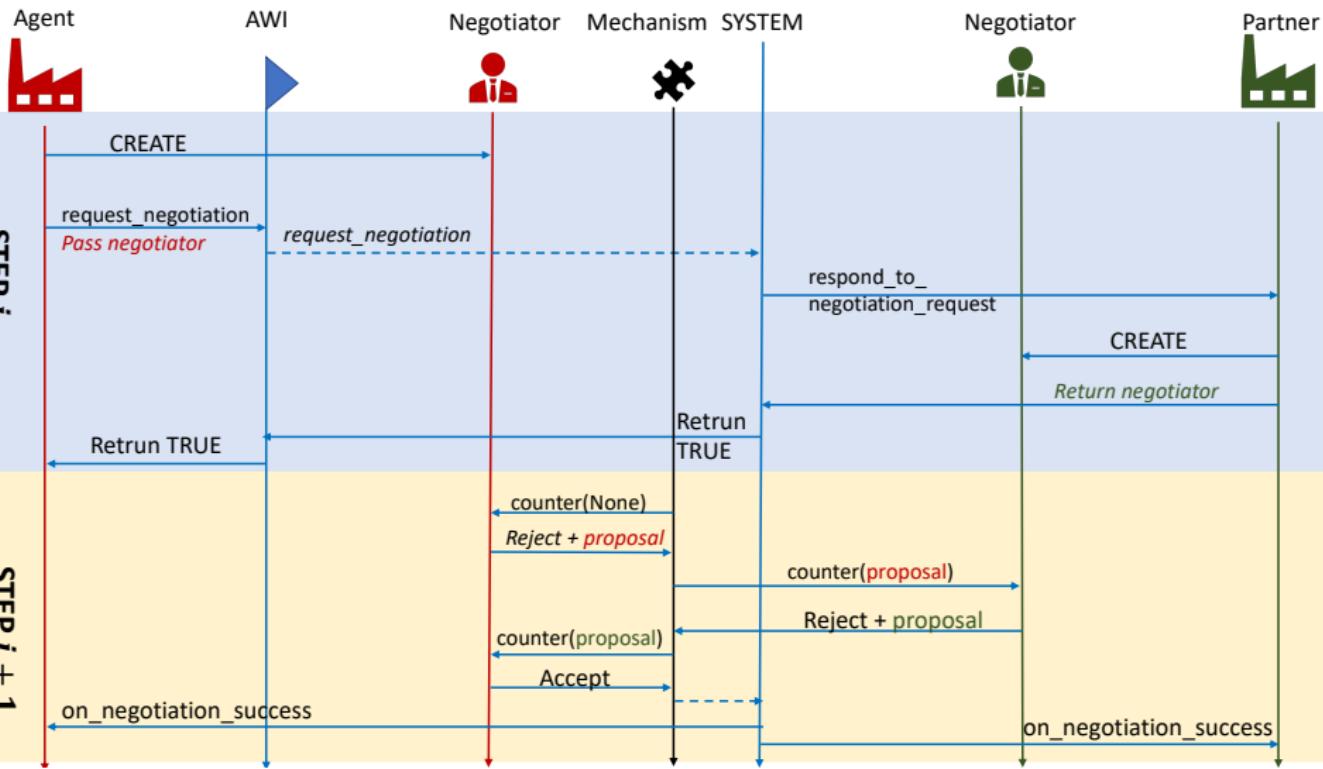
Callbacks and Timing



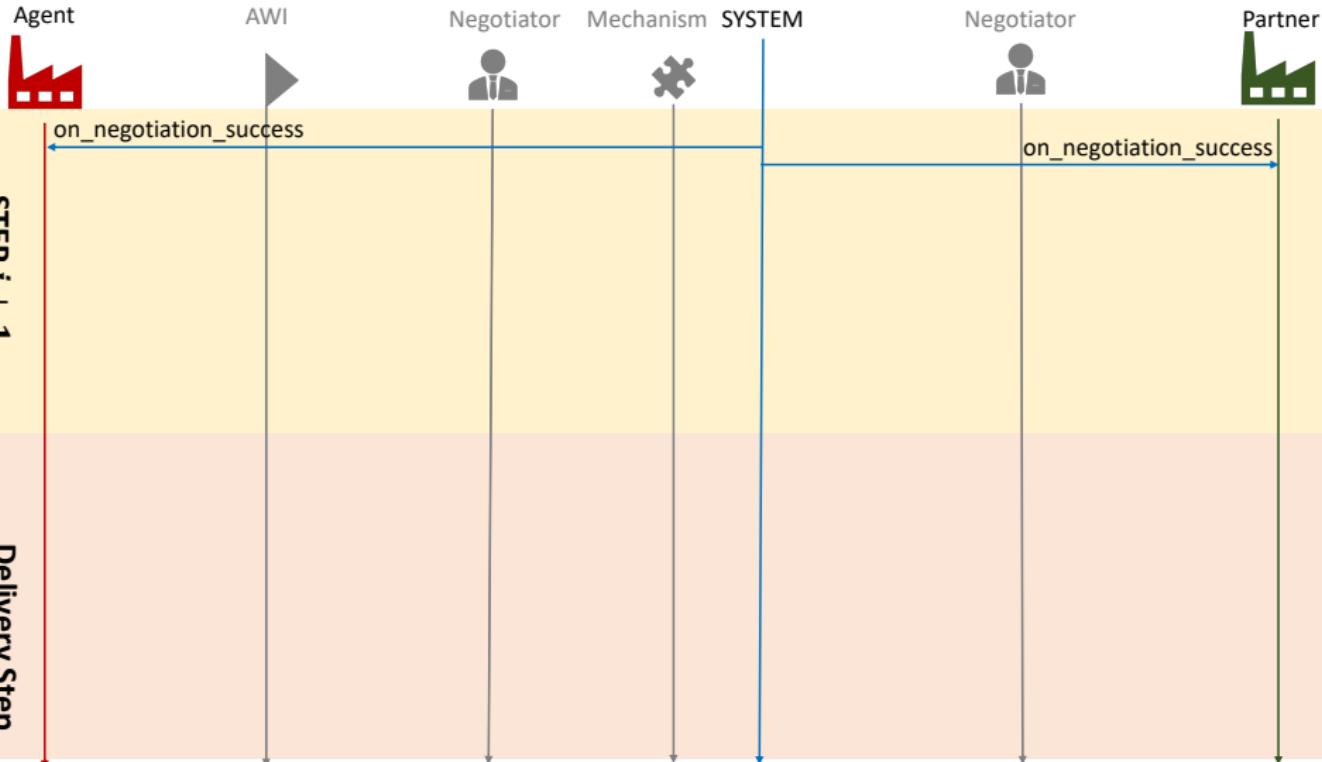
Callbacks and Timing



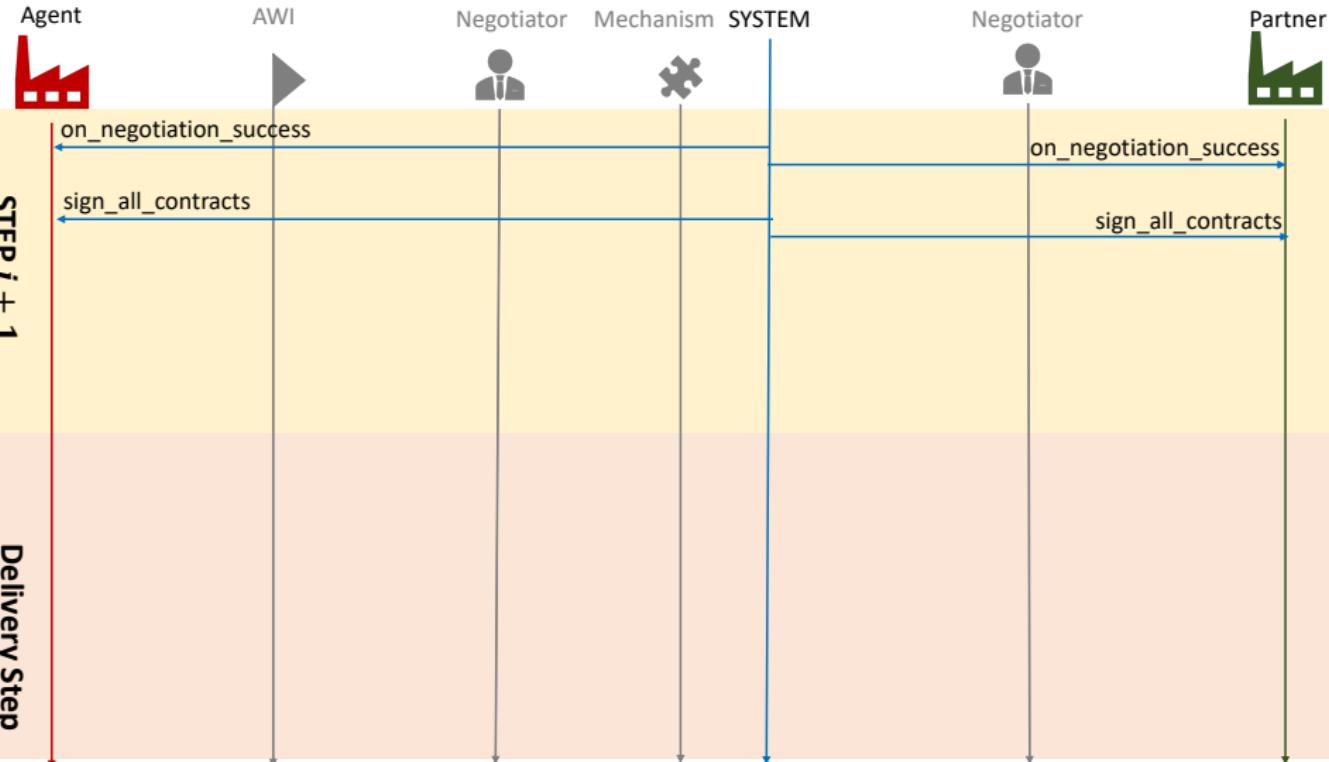
Callbacks and Timing



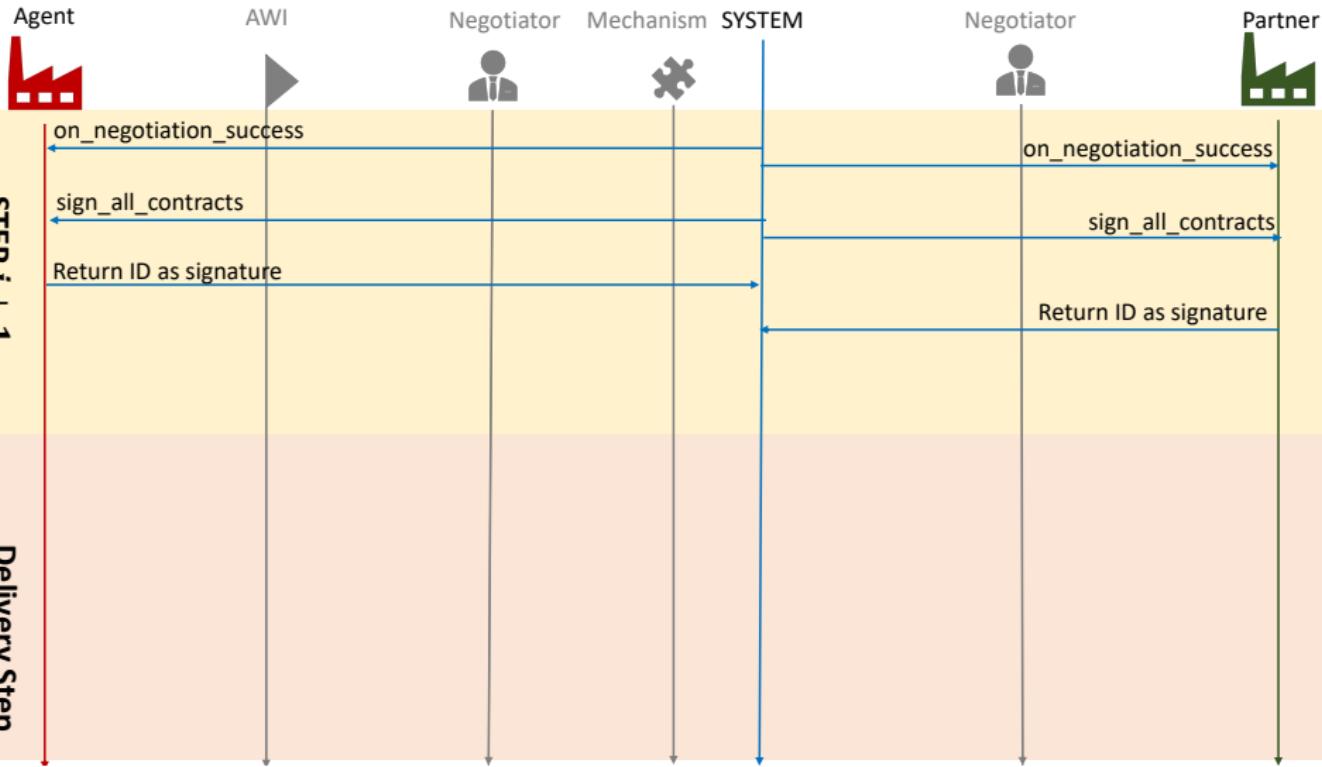
Callbacks and Timing



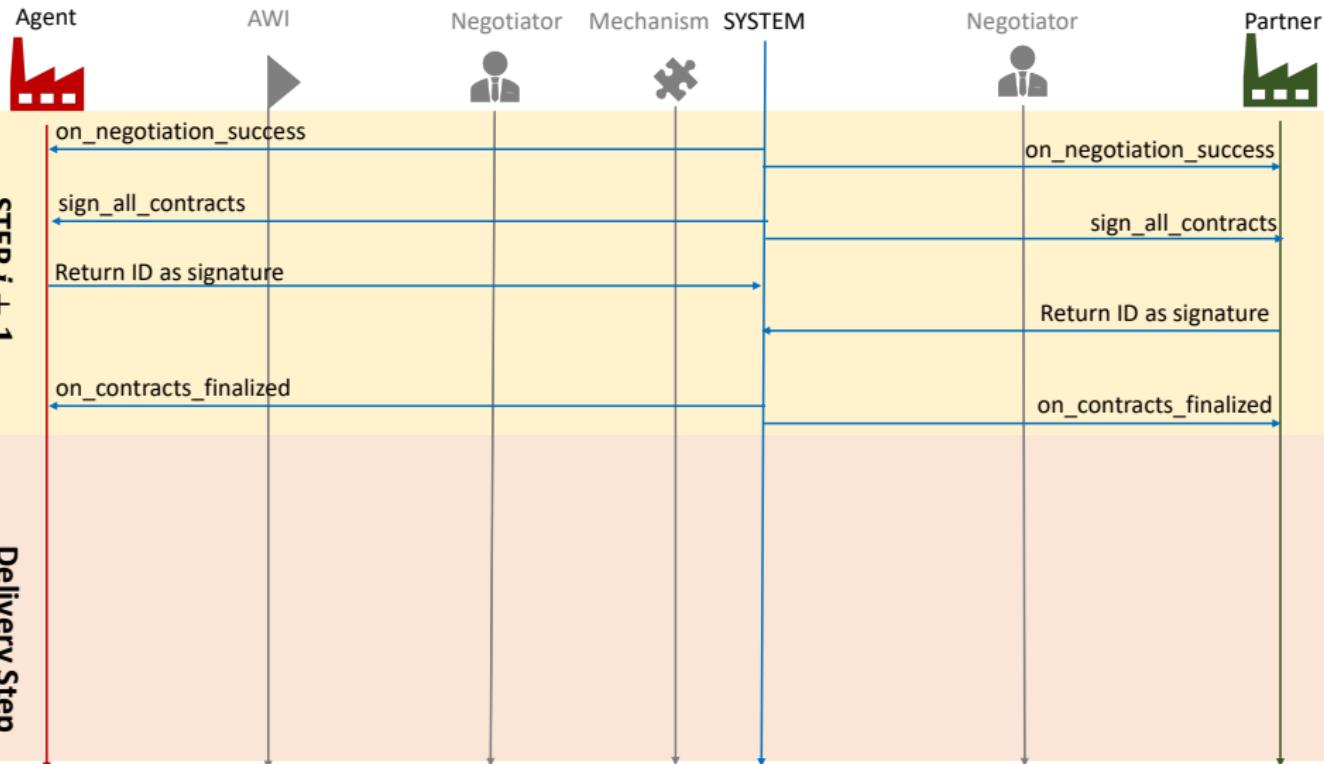
Callbacks and Timing



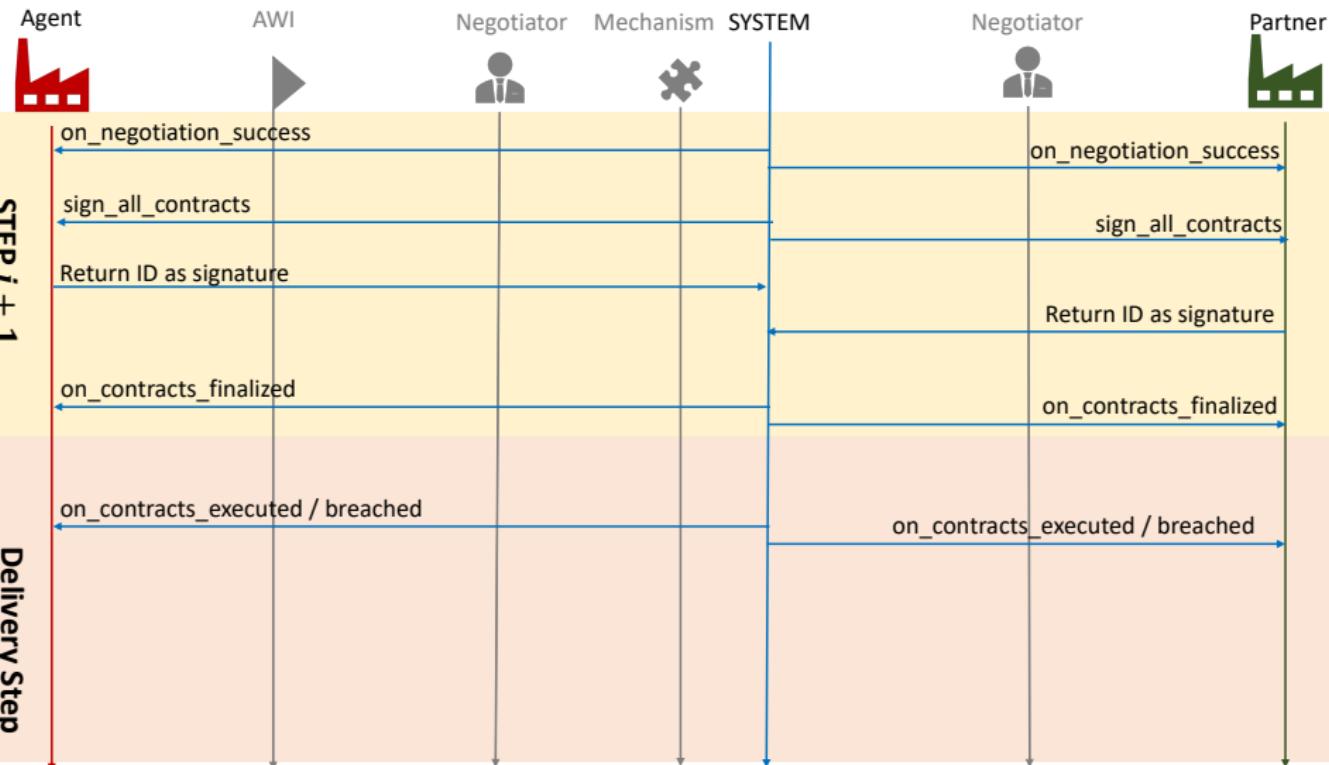
Callbacks and Timing



Callbacks and Timing



Callbacks and Timing



Callbacks and Timing

Agent

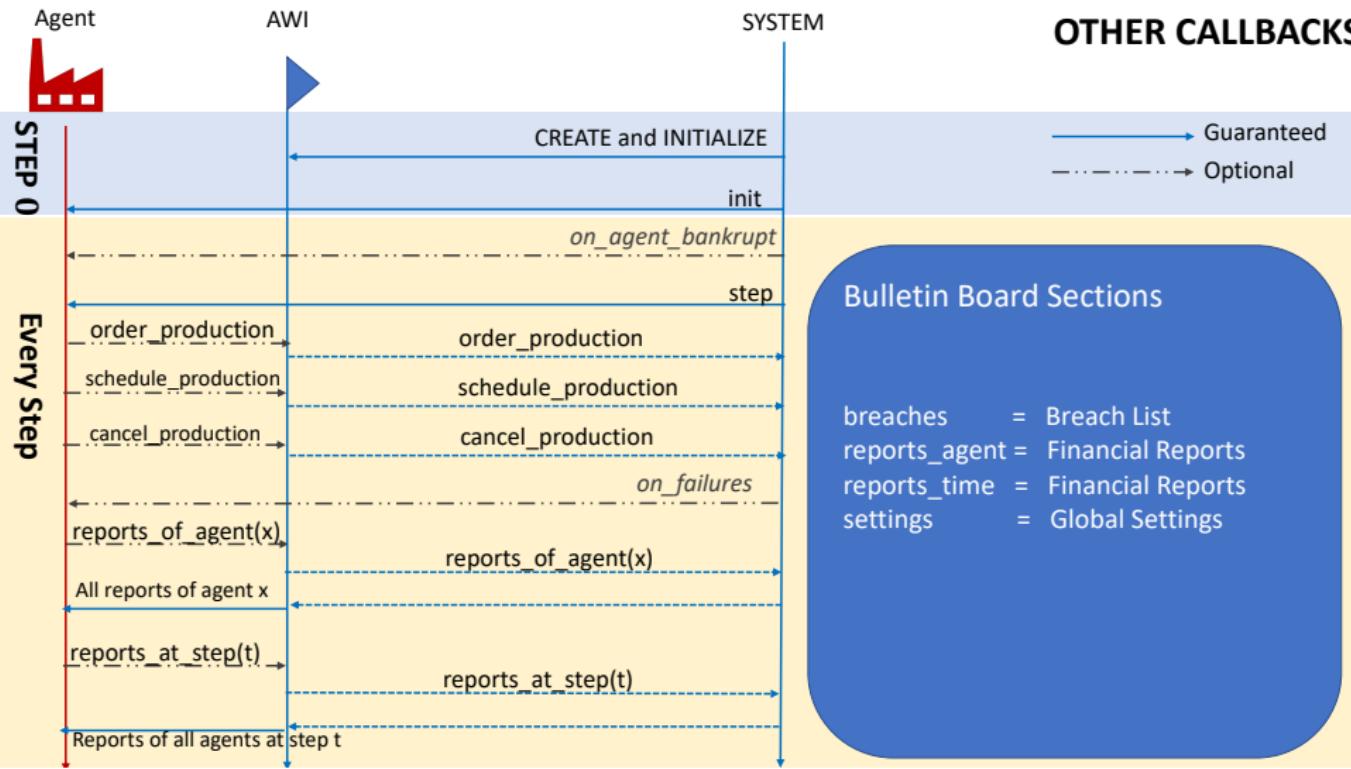


AWI



SYSTEM

OTHER CALLBACKS



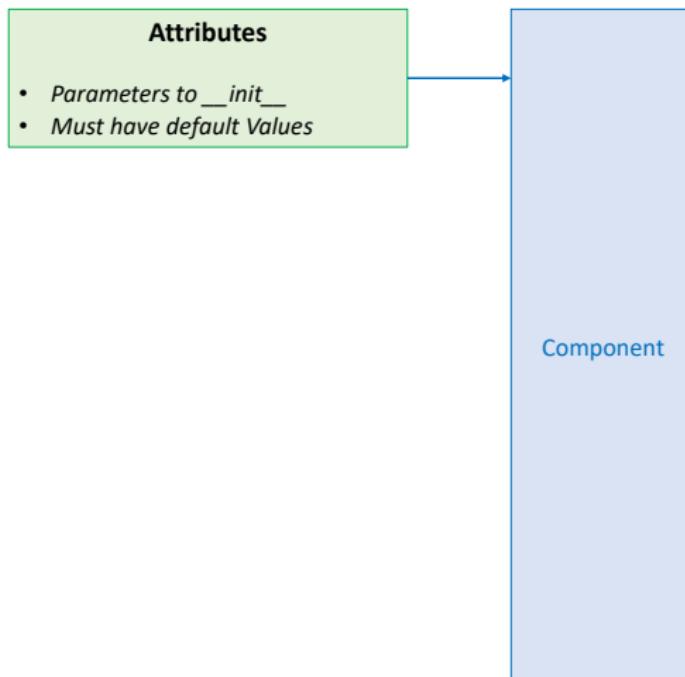
Bulletin Board Sections

breaches = Breach List
 reports_agent = Financial Reports
 reports_time = Financial Reports
 settings = Global Settings

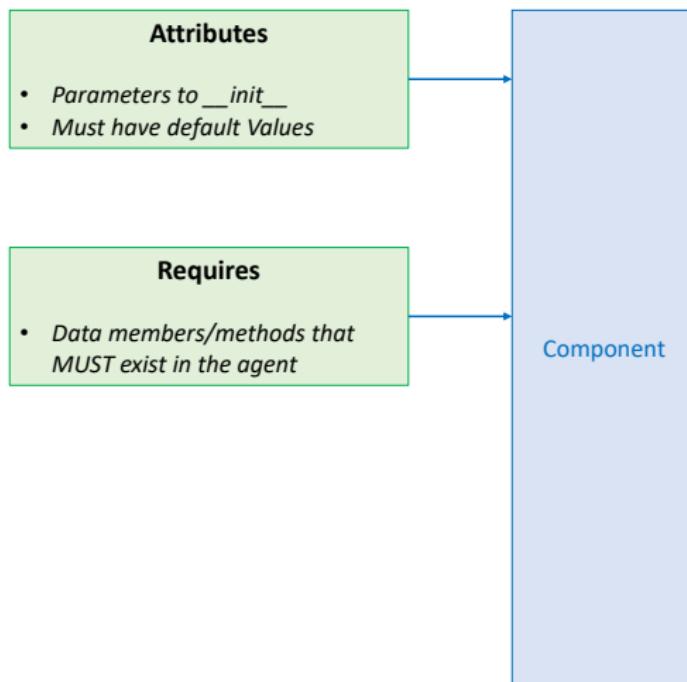
What is a component?



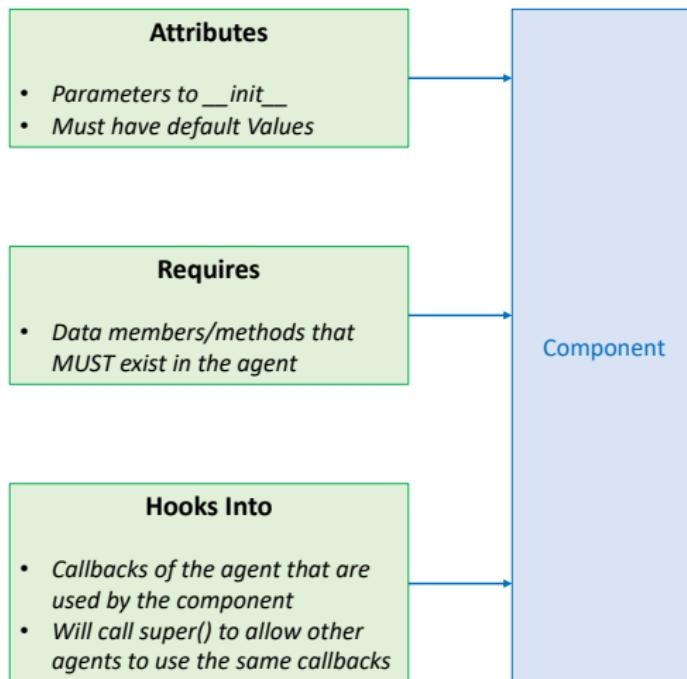
What is a component?



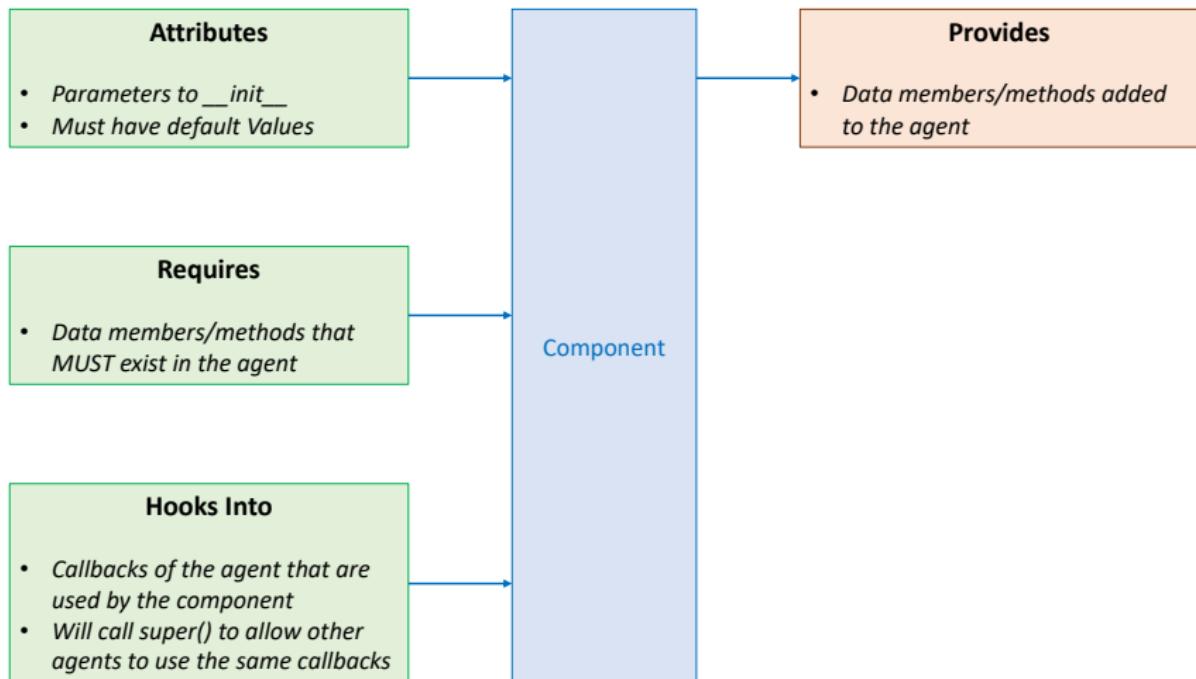
What is a component?



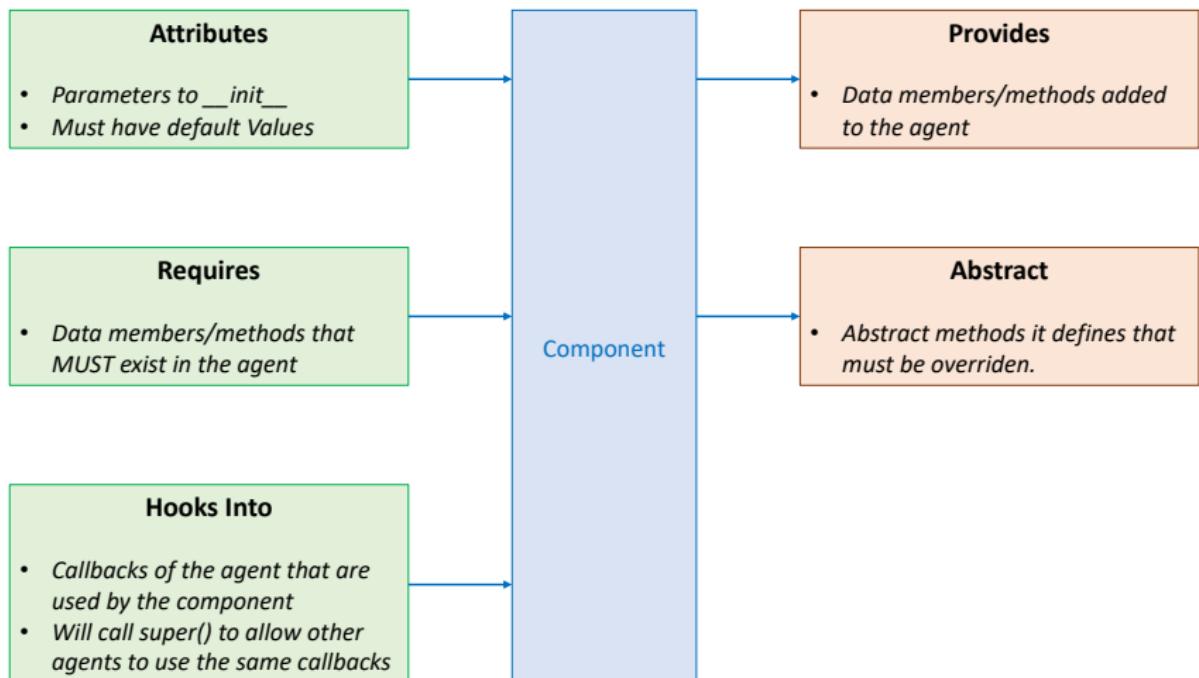
What is a component?



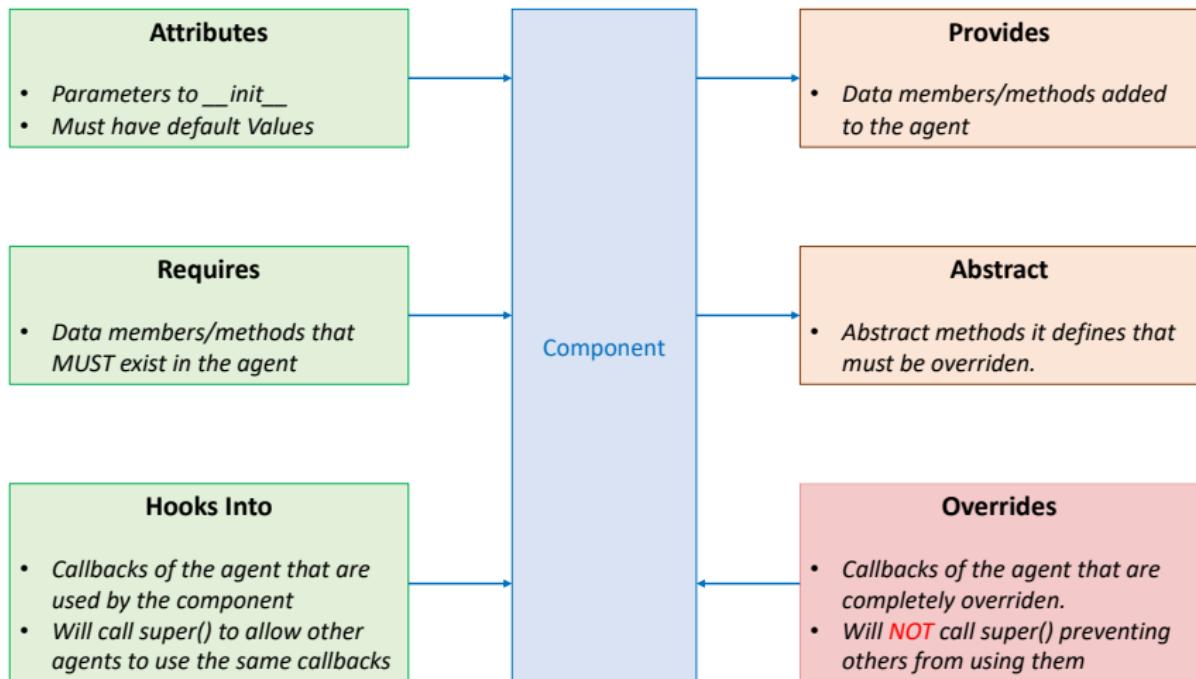
What is a component?



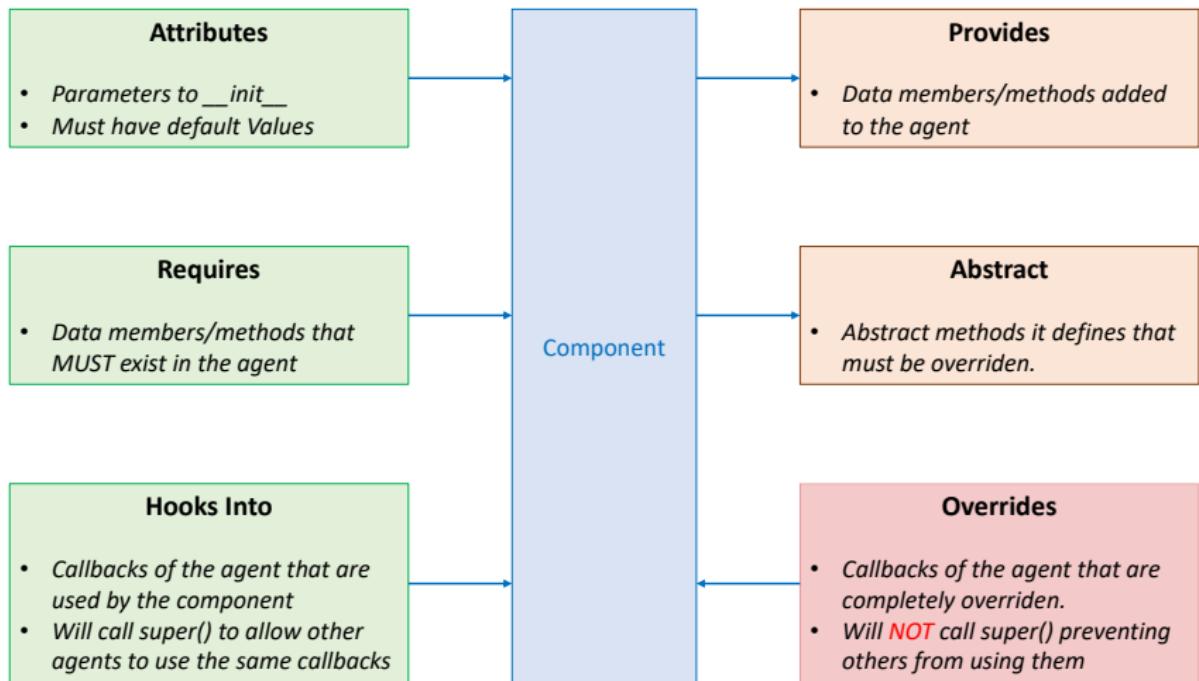
What is a component?



What is a component?

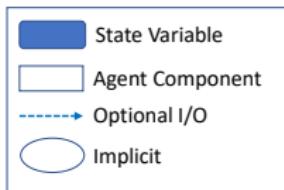


What is a component?



Hook into step, init, and on_ and override anything else*

SCML Agent Components

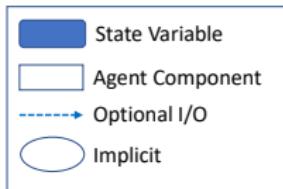


Production Strategy

Trading Strategy

Negotiation Manager

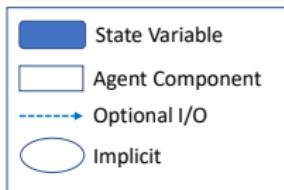
SCML Agent Components



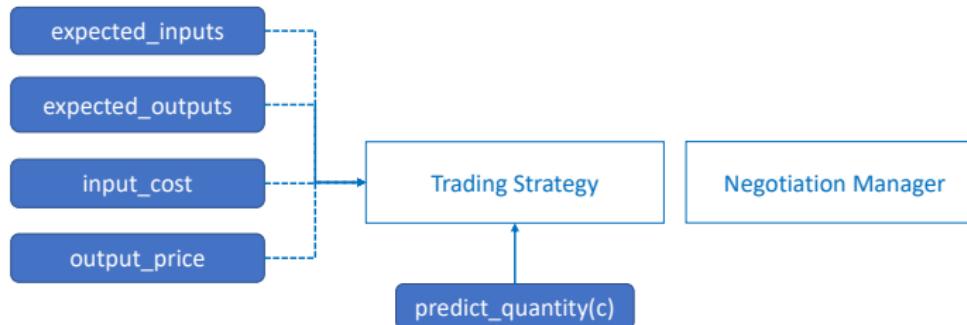
Production Strategy



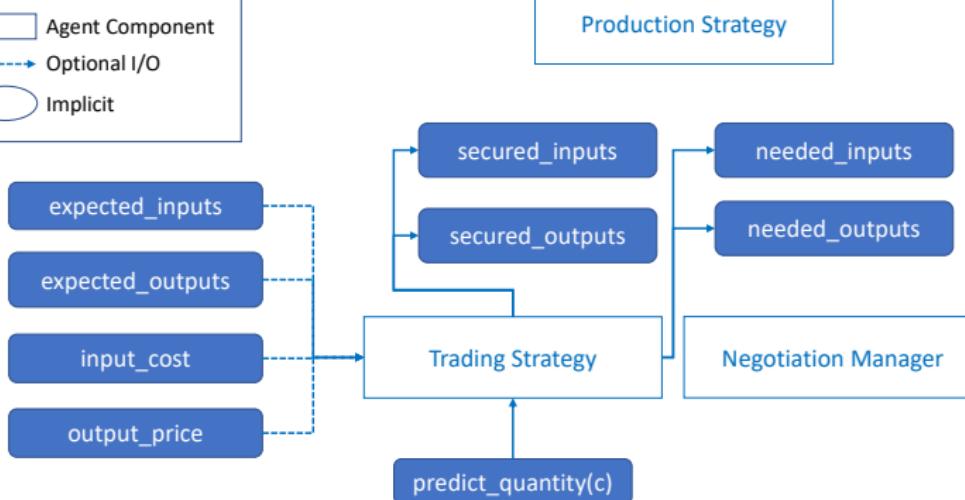
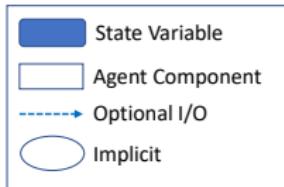
SCML Agent Components



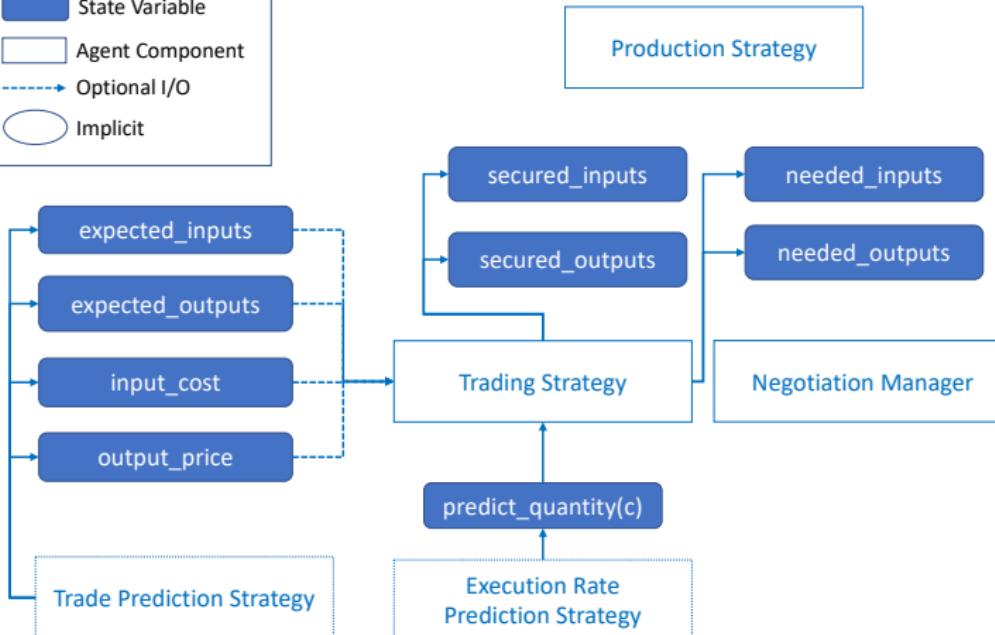
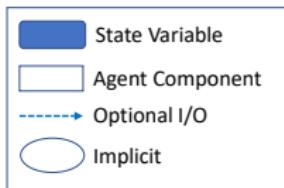
Production Strategy



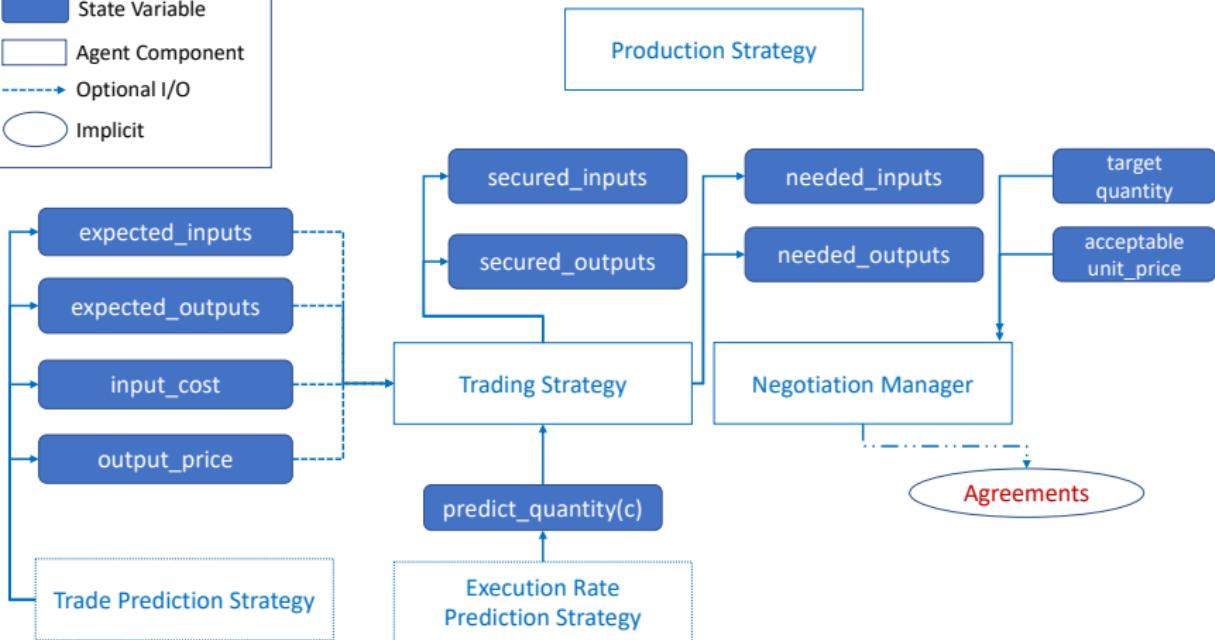
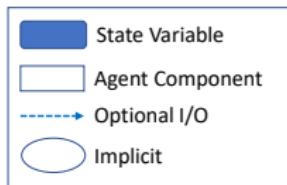
SCML Agent Components



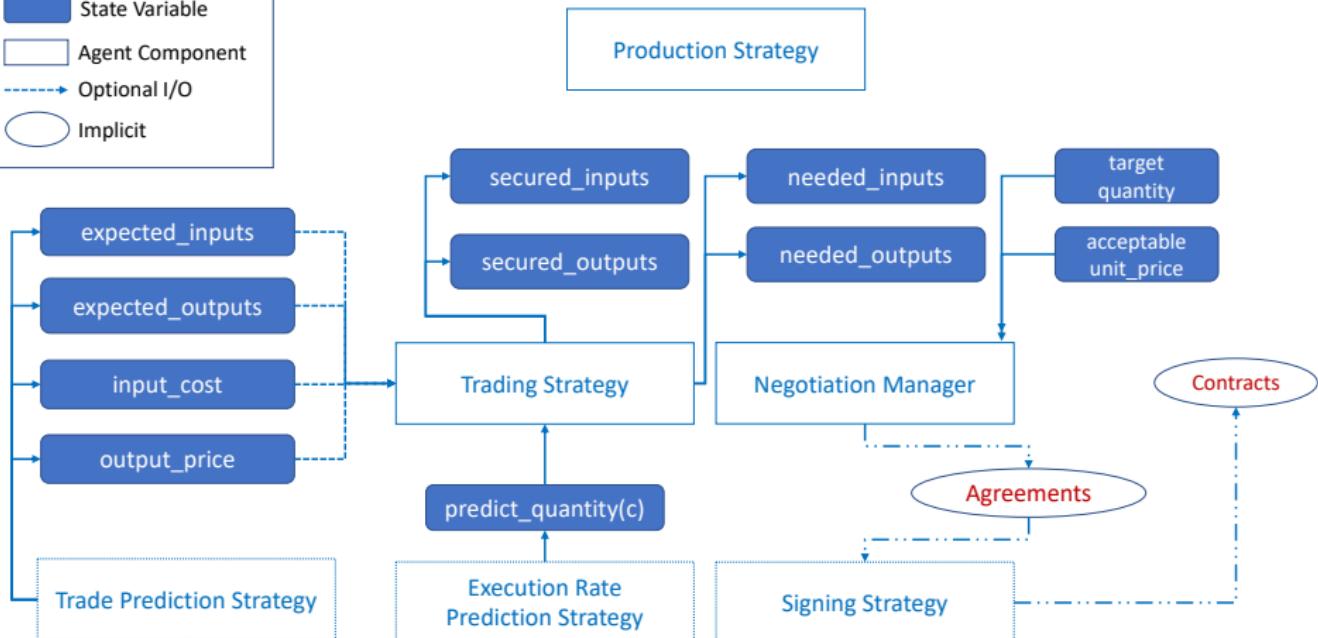
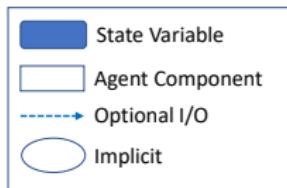
SCML Agent Components



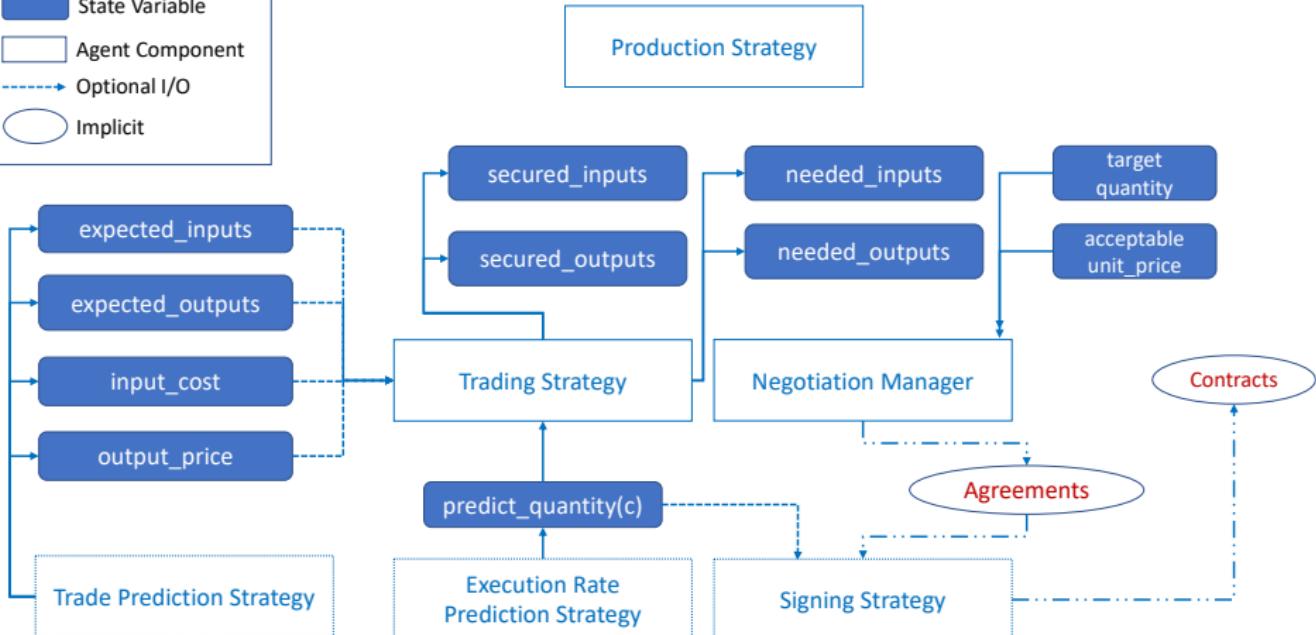
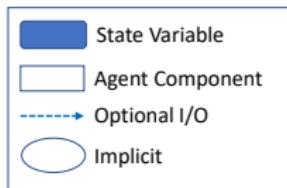
SCML Agent Components



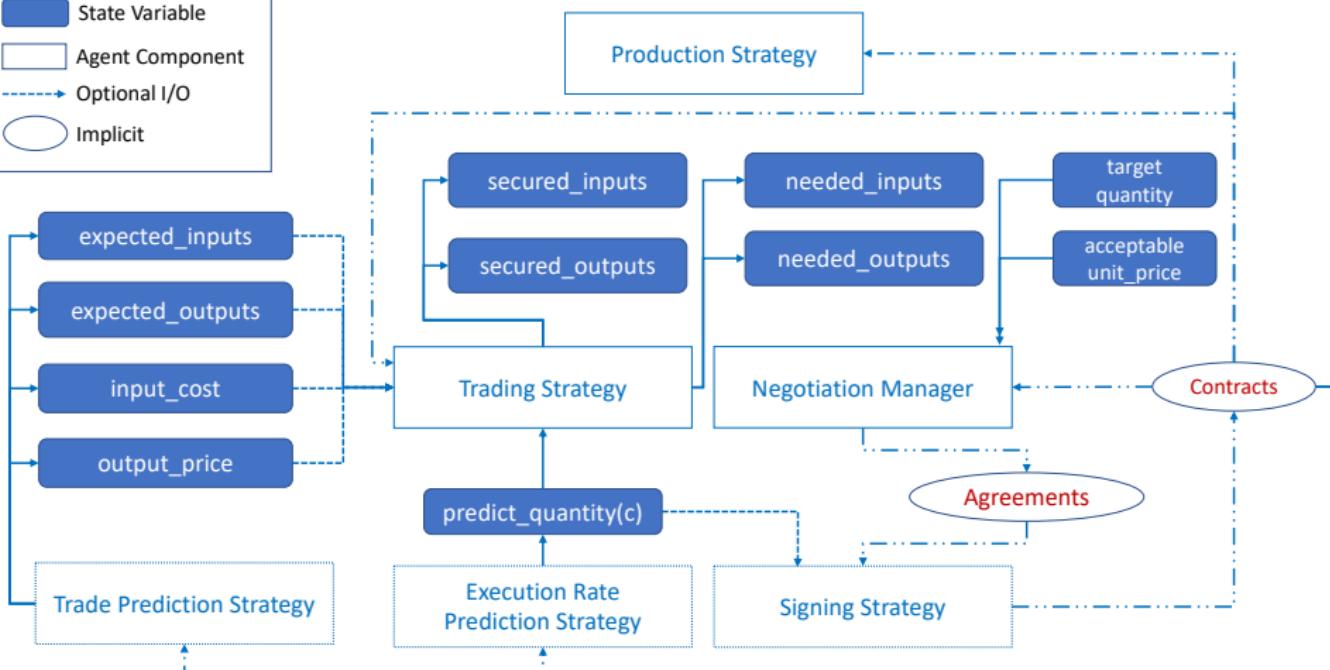
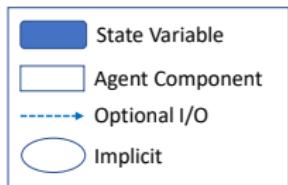
SCML Agent Components



SCML Agent Components



SCML Agent Components



Trading Strategy

Role



- Decides an overall **business plan**.
- Keeps track of buy/sell **needs**.

Built-in Options

No Strategy yes, just do nothing.

Reactive Zero needs.

Prediction Based trade and execution prediction.

- Needs come from trade predictions.
- Secured inputs/outputs come from execution prediction.

Negotiation Manager

Role

Negotiation Manager

target_quantity

For example, Needed – Secured

acceptable_unit_price

For example, catalog_price

Controls negotiation.

- Sets negotiation agendas → **Proactive** .
- Accepts/rejects negotiation requests → **Reactive** .
- Defines utility functions.
- Goal: Achieve the target put by the trading strategy.

Built-in Options

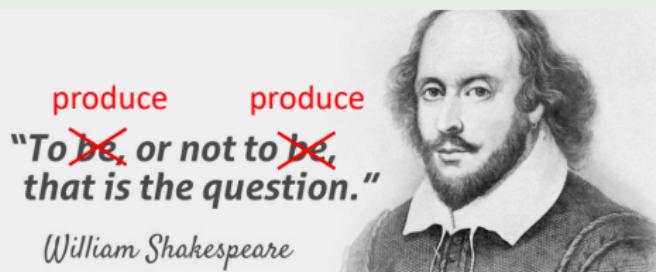
Independent Negotiations buy cheap ASAP, sell expensive ALAP.

Moving Range Creates one controller for selling and another for buying.

Step Manager Creates one controller per simulation step.

Production Strategy

Role



Built-in Options

Supply Driven Produce based on **buy** contracts.

- Inventory is always valued at the end.

Demand Driven Produce based on **sell** contracts.

- ... but inventory is valued at half the *trading* price!

Outline

1 Why Negotiation + SCM?

2 World Description

3 Agent

4 ANAC: A brief history

5 References

A brief history of ANAC

- The Automated Negotiating Agents Competition (ANAC)
- Started on 2010 and is currently in its 11th incarnation.
- Was conducted in conjunction with AAMAS but currently IJCAI.



Year	Challenge	Year	Challenge
2010	Domain Independence	2011	Linear Ufuns
2012	Reservation Value	2014	Learning and Adaptation
2015	Three-party negotiation	2016	Energy Grid Theme
2017	Repeated Negotiations, Diplomacy, HAN	2018	Repeated Negotiations, Diplomacy, HAN
2019	Elicitation, Diplomacy, HAN, Werewolf, SCML	2020	Uncertainty, HAN, Were-wolf, SCML , HUMAINE

Outline

1 Why Negotiation + SCM?

2 World Description

3 Agent

4 ANAC: A brief history

5 References

References I

Mohammad, Y., Viqueira, E. A., Ayerza, N. A., Greenwald, A., Nakadai, S., and Morinaga, S. (2019). Supply chain management world. In *International Conference on Principles and Practice of Multi-Agent Systems*, pages 153–169. Springer.

Automated Negotiation: Challenges and Tools

Future Challenges and Open Problems

Yasser Mohammad^{1,2,3} and Amy Greenwald⁴

¹ NEC CORPORATION, Global Innovation Unit

² National Institute of Advanced Industrial Science and Technology (AIST)

³ Assiut University, Egypt

⁴ Brown University

February 23rd, 2022

Outline

1 Situated Negotiation

- Negotiation Under Uncertainty

2 Optimality Results

- Against Known Opponents
- In Specific Settings

3 Preference Elicitation During Negotiation

- Procedure and Strategies
- Optimal Elicitation Algorithm
- Value of Information Algorithm

4 Concurrent Negotiation

5 References

Outline

1 Situated Negotiation

- Negotiation Under Uncertainty

2 Optimality Results

3 Preference Elicitation During Negotiation

4 Concurrent Negotiation

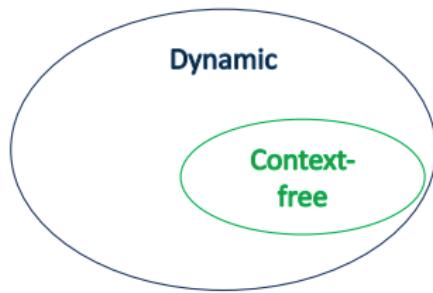
5 References

Situated Negotiation

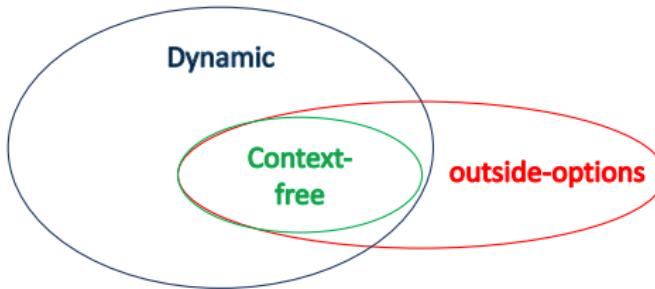


Context-free

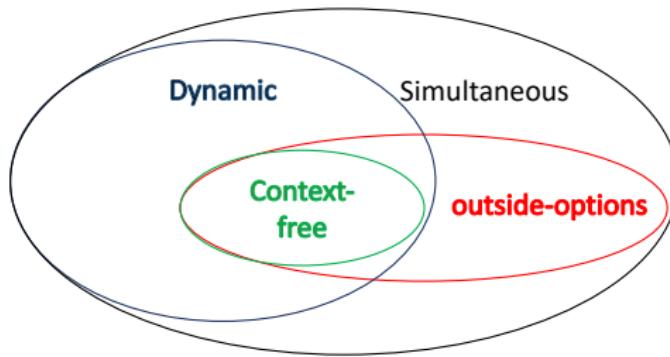
Situated Negotiation



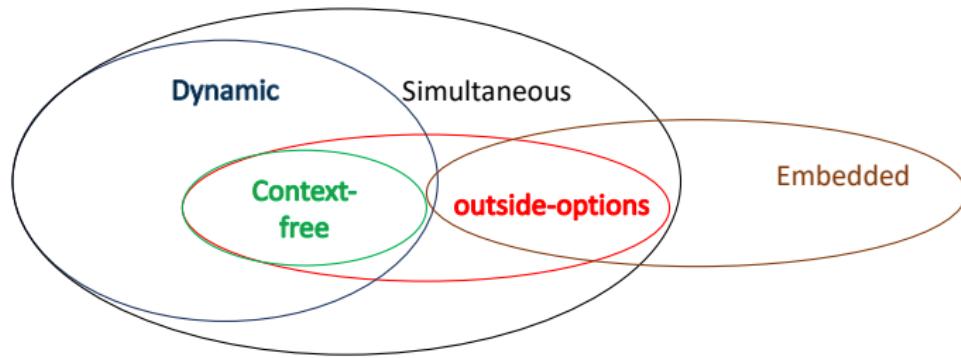
Situated Negotiation



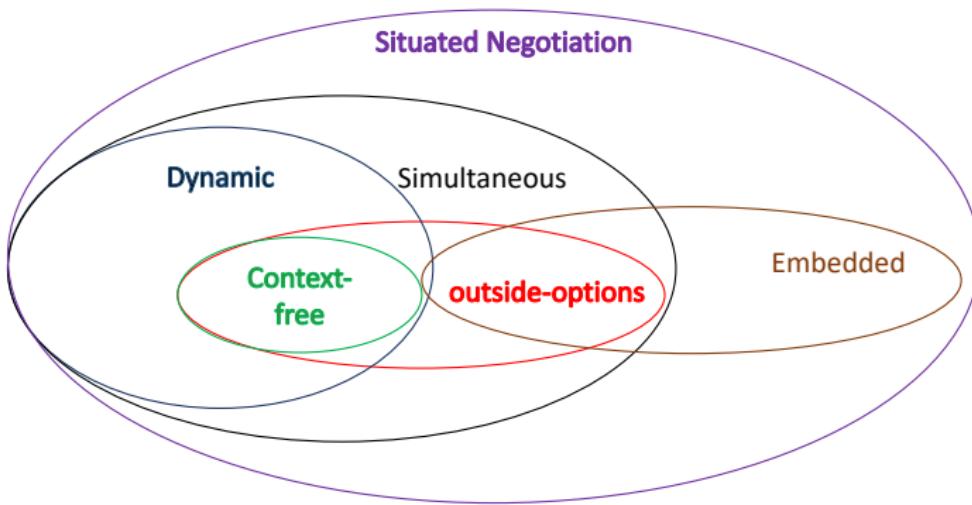
Situated Negotiation



Situated Negotiation



Situated Negotiation



Outline

1 Situated Negotiation

- Negotiation Under Uncertainty

2 Optimality Results

3 Preference Elicitation During Negotiation

4 Concurrent Negotiation

5 References

Negotiation Under Uncertainty

The challenge

How to negotiate when you have *partial* information about your actor's utility function?

Negotiation Under Uncertainty

The challenge

How to negotiate when you have *partial* information about your actor's utility function?

The Game

- ANAC 2019 agent game @ IJCAI introduced the first competition in this domain.
- Input is a ranking of a subset of the outcomes.

Outline

1 Situated Negotiation

2 Optimality Results

- Against Known Opponents
- In Specific Settings

3 Preference Elicitation During Negotiation

4 Concurrent Negotiation

5 References

Outline

1 Situated Negotiation

2 Optimality Results

- Against Known Opponents
- In Specific Settings

3 Preference Elicitation During Negotiation

4 Concurrent Negotiation

5 References

Outline

1 Situated Negotiation

2 Optimality Results

- Against Known Opponents
- In Specific Settings

3 Preference Elicitation During Negotiation

4 Concurrent Negotiation

5 References

Outline

1 Situated Negotiation

2 Optimality Results

3 Preference Elicitation During Negotiation

- Procedure and Strategies
- Optimal Elicitation Algorithm
- Value of Information Algorithm

4 Concurrent Negotiation

5 References

Outline

- 1 Situated Negotiation
- 2 Optimality Results
- 3 Preference Elicitation During Negotiation
 - Procedure and Strategies
 - Optimal Elicitation Algorithm
 - Value of Information Algorithm
- 4 Concurrent Negotiation
- 5 References

Preference Elicitation

The challenge

How to reduce Uncertainty in user preferences:

- before negotiation (offline preference elicitation).
- while negotiating (online preference elicitation).

Preference Elicitation

The challenge

How to reduce Uncertainty in user preferences:

- before negotiation (offline preference elicitation).
- while negotiating (online preference elicitation).

Types of questions

Utility Value what is $\tilde{u}(\omega)$?

Utility Constraint Is $\tilde{u}(\omega) \geq x$? Usually implemented as a standard gamble.

Utility Comparison Is $\omega_1 \succ \omega_2$?

Elicitation Procedures

- ① Long history in the decision support and economics research community.
- ② Take away message: .

Elicitation Procedures

- ① Long history in the decision support and economics research community.
- ② Take away message: **Do not ask about the utility directly..**

Elicitation Procedures

- ① Long history in the decision support and economics research community.
- ② Take away message: **Do not ask about the utility directly..**
- ③ Practical elicitation uses a **series** of comparisons between outcomes to assess utilities.

Elicitation Procedures

- ① Long history in the decision support and economics research community.
- ② Take away message: **Do not ask about the utility directly..**
- ③ Practical elicitation uses a **series** of comparisons between outcomes to assess utilities.

A Gamble

(ω^*, ω_*, p) : Getting ω^* with probability p otherwise ω_*

Elicitation Procedures

- ① Long history in the decision support and economics research community.
- ② Take away message: **Do not ask about the utility directly..**
- ③ Practical elicitation uses a **series** of comparisons between outcomes to assess utilities.

A Gamble

(ω^*, ω_*, p) : Getting ω^* with probability p otherwise ω_*

Example query

Do you prefer to get ω for certain over (ω^*, ω_*, p) ?

Elicitation Procedures/Strategies

Probability Equivalence

find p so that $\omega = (\omega^*, \omega_*, p)$

Certainty Equivalence

find ω so that $\omega = (\omega^*, \omega_*, p)$

- Both require *normalized* utilities.
- Both require knowledge of $\omega^* \succ \omega \succ \omega_*$.
- Lead to different biases.

Comparison-only Procedures

① Titration-down: $p_k = 1 - s \times k$

② Titration-up: $p_k = s \times k$

③ Ping-pong: $p_k = \begin{cases} s \times \lfloor k/2 \rfloor & k \text{ is odd} \\ 1 - s \times k/2 & k \text{ is even} \end{cases}$

Importance of Elicitation

Negotiation with Elicitation

$$m, \Omega, R, \tilde{U}_i \forall 1 \leq i \leq m, \hat{U}_i^0 \forall 1 \leq i \leq m$$

m Number of agents/actors

$\Omega = \{\omega_j\}$ Possible outcomes (assumed countable)

n Number of outcomes $|\Omega|$

$R(i) \equiv r_i$ Reserved value for agent *i*

$\tilde{U}_i : \Omega \rightarrow [0, 1]$ Utility of outcomes to **actor *i***

$\hat{U}_i^0 : \Omega \rightarrow P$ Probability distribution of utility values for **agent *i***

$$\hat{U}_{ij}^0 \equiv \hat{U}_i^0(\omega_j)$$

$P : \{[0, 1] \rightarrow [0, 1]\}$ A probability distribution on the closed interval $[0, 1]$

What is Elicitation Doing?

Reduces uncertainty in \hat{U}

State of the Art

- Lots of work on preferences/utility elicitation in decision making domain.
- Some work on incremental utility elicitation.
- Few works on incremental utility elicitation during negotiations

State of the Art

- Lots of work on preferences/utility elicitation in decision making domain.
- Some work on incremental utility elicitation.
- Few works on incremental utility elicitation during negotiations

Why Is Negotiation Different

- ① The acceptance model changes over time → environment dynamics are not static.
- ② Exploration is extremely costly.
- ③ Usually negotiations are not repeated much.
- ④ Cannot train on a simulator (in most cases).

Outline

1 Situated Negotiation

2 Optimality Results

3 Preference Elicitation During Negotiation

- Procedure and Strategies
- Optimal Elicitation Algorithm
- Value of Information Algorithm

4 Concurrent Negotiation

5 References

Pandora's Problem [Economics]

- ❶ A set of n boxes ($\{\omega_j\}$).
- ❷ Opening a box j gives a reward between 0 and ∞ according to distribution p_j after t_j time-steps, and costs c_j .
- ❸ Future rewards are discounted with a known factor β .



Pandora's Problem [Economics]

- ❶ A set of n boxes ($\{\omega_j\}$).
- ❷ Opening a box j gives a reward between 0 and ∞ according to distribution p_j after t_j time-steps, and costs c_j .
- ❸ Future rewards are discounted with a known factor β .



Pandora's Problem [Economics]

- ① A set of n boxes ($\{\omega_j\}$).
- ② Opening a box j gives a reward between 0 and ∞ according to distribution p_j after t_j time-steps, and costs c_j .
- ③ Future rewards are discounted with a known factor β .
- ④ Pandora's Problem:
 - ① What is the optimal order to open the boxes?
 - ② When should she stop?



Pandora's Problem [Economics]

- ① A set of n boxes ($\{\omega_j\}$).
- ② Opening a box j gives a reward between 0 and ∞ according to distribution p_j after t_j time-steps, and costs c_j .
- ③ Future rewards are discounted with a known factor β .
- ④ Pandora's Problem:
 - ① What is the optimal order to open the boxes?
 - ② When should she stop?
- ⑤ Similar to elicitation (boxes = outcomes, open = query) but assumes that uncertainty is completely removed.



Pandora's Problem [Economics]

- ① A set of n boxes ($\{\omega_j\}$).
- ② Opening a box j gives a reward between 0 and ∞ according to distribution p_j after t_j time-steps, and costs c_j .
- ③ Future rewards are discounted with a known factor β .
- ④ Pandora's Problem:
 - ① What is the optimal order to open the boxes?
 - ② When should she stop?
- ⑤ Similar to elicitation (boxes = outcomes, open = query) but assumes that uncertainty is completely removed.



Pandora's Problem [Economics]

- ① A set of n boxes ($\{\omega_j\}$).
- ② Opening a box j gives a reward between 0 and ∞ according to distribution p_j after t_j time-steps, and costs c_j .
- ③ Future rewards are discounted with a known factor β .
- ④ Pandora's Problem:
 - ① What is the optimal order to open the boxes?
 - ② When should she stop?
- ⑤ Similar to elicitation (boxes = outcomes, open = query) but assumes that uncertainty is completely removed.

Solution: Pandora's Rule [Weitzman, 1979]

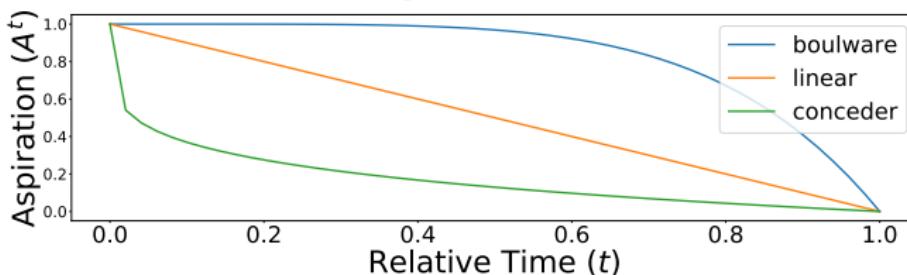
For each box j , find z_j which is the solution to:

$$c_j = \beta_j \int_{z_j}^{\infty} (u - z_j) p_j(u) du - (1 - \beta_j) z_j$$

Open the box with highest (z_i) if it is larger than the best utility.

Optimal Elicitation [Baarslag and Gerdin, 2015]

Adapts Pandora's Rule to the negotiation context:



- ① $\beta = 1.0$
- ② Define aspiration level as: $A^t \equiv r_i + (A^0 - r_i) \times \left(1 - \frac{t}{N}^{1/e}\right)$
 $e > 1 \rightarrow$ Boulware, $e = 1 \rightarrow$ Linear, $e < 1 \rightarrow$ Conceder
- ③ $p_j = \Lambda_i^t(\omega_j) \times \mathbb{E}(\hat{U}_{ij}^t) + (1 - \Lambda_i^t(\omega_j)) \times A^t(\omega_j)$
- ④ Assume that there is an open box giving r_i with outcome index 0.
- ⑤ End the negotiation once the best box is 0.

Why is OE sub-optimal?

Main Issue

Assuming that all uncertainty is removed by elicitation.

- ➊ Assuming that $\hat{U}_{ij} \rightarrow \delta \left[u = \tilde{U}_i(\omega_j) \right]$

Why is OE sub-optimal?

Main Issue

Assuming that all uncertainty is removed by elicitation.

- ① Assuming that $\hat{U}_{ij} \rightarrow \delta \left[u = \tilde{U}_i(\omega_j) \right]$
- ② Consider any practical strategy (e.g. titration-down):
 - After the first question: $\hat{U}_{ij}^t \rightarrow \hat{U}_{ij}^{t+1}$
 - z_j was calculated using \hat{U}_{ij}^t and must be recalculated.

Take-away message

Avoid deep-elicitation.

Extensions to Pandora's algorithm

Closed-form Calculation of z-index[Mohammad and Nakadai, 2018b]

$$z_j = \begin{cases} \frac{a+b}{2}\beta - c_j & z_j \leq a \\ \frac{-\lambda \pm \sqrt{\lambda^2 - 4\zeta}}{2} & a < z_j \leq b \\ \lambda - 2 \left(b + \frac{a-\beta}{\beta}(b-a) \right) & \zeta = b^2 - \frac{2c_j}{\beta}(b-a) \end{cases}$$

The balanced expectation operator

$$\mathcal{E}(\hat{U}_{ij}^t) = \frac{t}{N} \times \text{Min} \left(\hat{U}_{ij}^t \right) + \left(1 - \frac{t}{N} \right) \times \text{Max} \left(\hat{U}_{ij}^t \right)$$

Min/Max a *biased estimator* that exaggerate the lower/upper part of its input. For $U(a, b)$, $\text{Min}, \text{Max} = a, b$.

Outline

1 Situated Negotiation

2 Optimality Results

3 Preference Elicitation During Negotiation

- Procedure and Strategies
- Optimal Elicitation Algorithm
- Value of Information Algorithm

4 Concurrent Negotiation

5 References

Value of Information Algorithm

- Based on [Chajewska et al., 2000] in decision-support context.
- Adapted to the negotiation context.

Main Idea

- Assume an accurate opponent model (acceptance probability)
- Given a set of queries $Q \rightarrow$ find the one with the maximum difference between the expected expected utility before and after asking it[Baarslag and Kaisers, 2017, Mohammad and Nakadai, 2018a].

VOI Based Elicitation

Policy

$$\pi^t = (\omega^t, \omega^{t+1}, \omega^N) \text{ where } \omega^x \in \Omega$$

$K(\omega|\pi)$ ≡ index of ω in π

$$\pi(k) = \omega \text{ where } K(\omega|\pi) = k$$

Optimal Policy

$$\pi^{t*} = \arg \max_{\pi} EEU^t \left(\pi, \left\{ \hat{U}_{\omega}^t \right\} \right)$$

VOI Based Elicitation

Policy

$$\pi^t = (\omega^t, \omega^{t+1}, \dots, \omega^N) \text{ where } \omega^x \in \Omega$$

$K(\omega|\pi)$ ≡ index of ω in π

$$\pi(k) = \omega \text{ where } K(\omega|\pi) = k$$

Probability of Agreement

$$Pa^t(\omega|\pi) = \begin{cases} \Lambda^t(\omega) \prod_{k=1}^{K_\pi(\omega)-1} (1 - \Lambda^t(\pi(k))) & \omega \in \pi \\ 0 & \text{otherwise} \end{cases}$$

Optimal Policy

$$\pi^{t*} = \arg \max_{\pi} EEU^t \left(\pi, \left\{ \hat{U}_{\omega}^t \right\} \right)$$

VOI Based Elicitation

Policy

$$\pi^t = (\omega^t, \omega^{t+1}, \dots, \omega^N) \text{ where } \omega^x \in \Omega$$

$K(\omega|\pi)$ ≡ index of ω in π

$$\pi(k) = \omega \text{ where } K(\omega|\pi) = k$$

Probability of Agreement

$$Pa^t(\omega|\pi) = \begin{cases} \Lambda^t(\omega) \prod_{k=1}^{K_\pi(\omega)-1} (1 - \Lambda^t(\pi(k))) & \omega \in \pi \\ 0 & \text{otherwise} \end{cases}$$

Expected Expected Utility [Boutilier, 2003]

$$EEU^t\left(\pi, \left\{ \hat{U}_\omega^t \right\}\right) = \sum_{\omega \in \Omega} Pa(\omega|\pi) \mathbb{E}\left(\hat{U}_\omega^t\right)$$

Optimal Policy

$$\pi^{t*} = \arg \max_{\pi} EEU^t\left(\pi, \left\{ \hat{U}_\omega^t \right\}\right)$$

VOI Based Elicitation II

Questions and Answers

$$\begin{aligned}Q &\equiv \{q_I\} \\ q_I &\equiv \{(Ans_s^I, p_s)\}\} \\ Ans_s^I &\equiv \{\hat{U}_\omega^{t+1}\} \\ \sum_s p_s &= 1\end{aligned}$$

VOI Based Elicitation II

Questions and Answers

$$\begin{aligned} Q &\equiv \{q_I\} \\ q_I &\equiv \{(Ans_s^I, p_s)\} \\ Ans_s^I &\equiv \{\hat{U}_\omega^{t+1}\} \\ \sum_s p_s &= 1 \end{aligned}$$

Expected value of information

$$EVOI(q^I, \{\hat{U}_\omega^t\}) = \mathbb{E}_s (\max_{\pi} EEU(\pi, Ans_s^I)) - \max_{\pi} EEU(\pi, \{\hat{U}_\omega^t\})$$

VOI Based Elicitation II

Questions and Answers

$$\begin{aligned}Q &\equiv \{q_I\} \\ q_I &\equiv \{(Ans_s^I, p_s)\} \\ Ans_s^I &\equiv \{\hat{U}_\omega^{t+1}\} \\ \sum_s p_s &= 1\end{aligned}$$

Expected value of information

$$EVOI(q^I, \{\hat{U}_\omega^t\}) = \mathbb{E}_s (\max_{\pi} EEU(\pi, Ans_s^I)) - \max_{\pi} EEU(\pi, \{\hat{U}_\omega^t\})$$

Elicitation

Ask q^* where

$$q^* = \arg \max_q (EVOI(q^I, \{\hat{U}_\omega^t\}) - c_q)$$

c_q Cost of asking question q

VOI main Issues

Accurate Agreement Model Assumption

- Everything depends on the probability of agreement (Pa)
- Pa depends on the **product** of probabilities in the acceptance model (Λ^t)

$$Pa^t(\omega|\pi) = \begin{cases} \Lambda^t(\omega) \prod_{k=1}^{K_\pi(\omega)-1} (1 - \Lambda^t(\pi(k))) & \omega \in \pi \\ 0 & \text{otherwise} \end{cases}$$

VOI main Issues

Accurate Agreement Model Assumption

- Everything depends on the probability of agreement (Pa)
- Pa depends on the **product** of probabilities in the acceptance model (Λ^t)

$$Pa^t(\omega|\pi) = \begin{cases} \Lambda^t(\omega) \prod_{k=1}^{K_\pi(\omega)-1} (1 - \Lambda^t(\pi(k))) & \omega \in \pi \\ 0 & \text{otherwise} \end{cases}$$

Speed: Complexity = $O(nN|Q||Ans|)$

- Too many argmax and \mathbb{E} operations.
- Every policy extends to the end of the negotiation.

$$q^* = \arg \max_q \left(EVOI \left(q', \left\{ \hat{U}_\omega^t \right\} \right) - c_q \right)$$

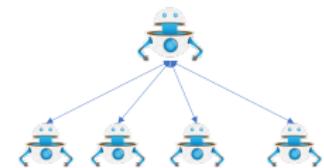
$$EVOI \left(q', \left\{ \hat{U}_\omega^t \right\} \right) = \mathbb{E}_s \left(\max_\pi EEU \left(\pi, Ans_s' \right) \right) - \max_\pi EEU \left(\pi, \left\{ \hat{U}_\omega^t \right\} \right)$$

$$\pi^{t*} = \arg \max_\pi EEU^t \left(\pi, \left\{ \hat{U}_\omega^t \right\} \right)$$

Outline

- 1 Situated Negotiation
- 2 Optimality Results
- 3 Preference Elicitation During Negotiation
- 4 Concurrent Negotiation
- 5 References

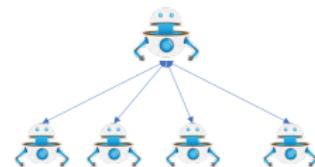
Concurrent Negotiation



Concurrent Negotiation

Generality

- Specific scenario (buyer-seller).
- General domain



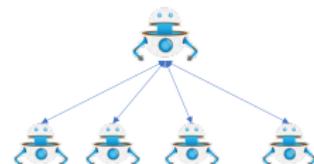
Concurrent Negotiation

Generality

- Specific scenario (buyer-seller).
- General domain

Decommitment

- Symmetric de-commitment.
- Asymmetric de-commitment.
- No de-commitment.



Concurrent Negotiation

Generality

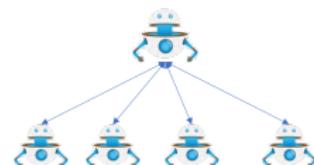
- Specific scenario (buyer-seller).
- General domain

Decommitment

- Symmetric de-commitment.
- Asymmetric de-commitment.
- No de-commitment.

Timing

- Synchronous.
- Any-time.



Outline

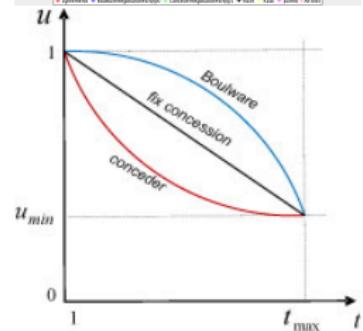
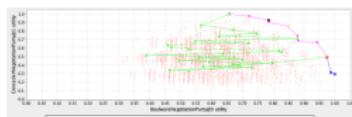
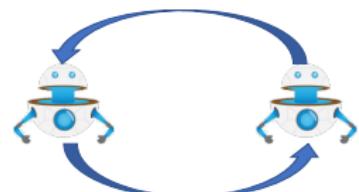
- 1 Situated Negotiation
- 2 Optimality Results
- 3 Preference Elicitation During Negotiation
- 4 Concurrent Negotiation
- 5 References

References I

- Baarslag, T. and Gerdin, E. H. (2015). Optimal incremental preference elicitation during negotiation. In *IJCAI*, pages 3–9.
- Baarslag, T. and Kaisers, M. (2017). The value of information in automated negotiation: A decision model for eliciting user preferences. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 391–400. International Foundation for Autonomous Agents and Multiagent Systems.
- Boutilier, C. (2003). On the foundations of expected expected utility. In *IJCAI*, volume 3, pages 285–290. Citeseer.
- Chajewska, U., Koller, D., and Parr, R. (2000). Making rational decisions using adaptive utility elicitation. In *AAAI/IAAI*, pages 363–369.
- Mohammad, Y. and Nakadai, S. (2018a). Fastvoi: Efficient utility elicitation during negotiations. In *International Conference on Principles and Practice of Multi-Agent Systems (PRIMA)*, pages 560–567. Springer.
- Mohammad, Y. and Nakadai, S. (2018b). Utility elicitation during negotiation with practical elicitation strategies. In *IEEE SMC*.
- Weitzman, M. L. (1979). Optimal search for the best alternative. *Econometrica: Journal of the Econometric Society*, pages 641–654.

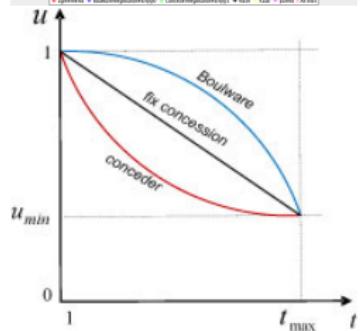
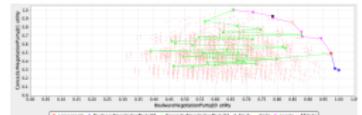
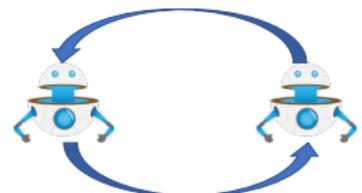
Conclusion

- Automated negotiation can enhance societal welfare.
- Genius and NegMAS as open-ended platforms for research in automated negotiation.
- Classical automated negotiation research in economics focused on simplified situations and provided performance guarantees.
- Many open questions:
 - General environment and unknown opponents.
 - Incomplete information about self.
 - Concurrent negotiations.
 - Negotiations with non-stationary utilities.
 - When to use negotiation?



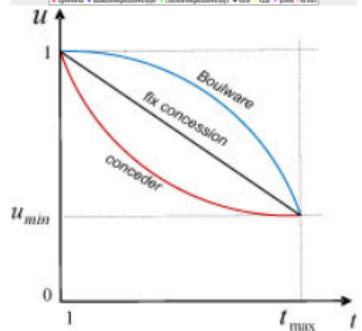
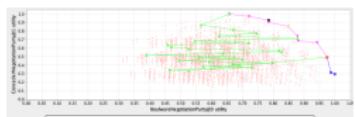
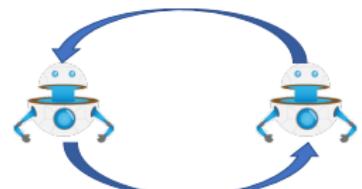
Conclusion

- Automated negotiation can enhance societal welfare.
- Genius and NegMAS as open-ended platforms for research in automated negotiation.
- Classical automated negotiation research in economics focused on simplified situations and provided performance guarantees.
- Many open questions:
 - General environment and unknown opponents.
 - Incomplete information about self.
 - Concurrent negotiations.
 - Negotiations with non-stationary utilities.
 - When to use negotiation?



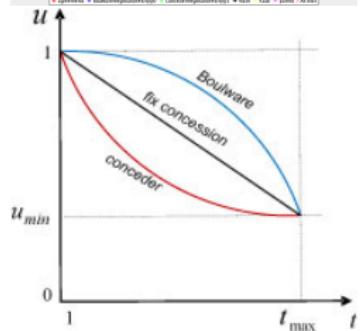
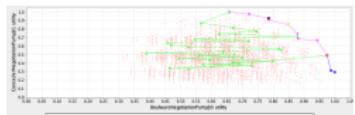
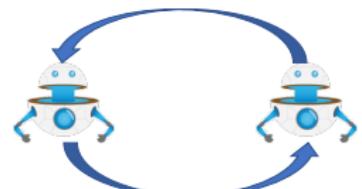
Conclusion

- Automated negotiation can enhance societal welfare.
- Genius and NegMAS as open-ended platforms for research in automated negotiation.
- Classical automated negotiation research in economics focused on simplified situations and provided performance guarantees.
- Many open questions:
 - General environment and unknown opponents.
 - Incomplete information about self.
 - Concurrent negotiations.
 - Negotiations with non-stationary utilities.
 - When to use negotiation?



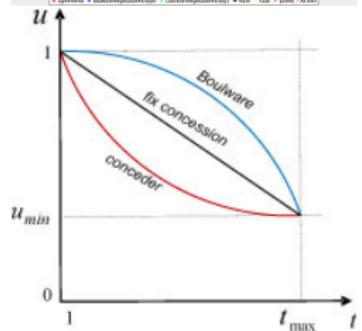
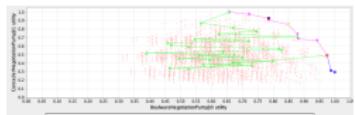
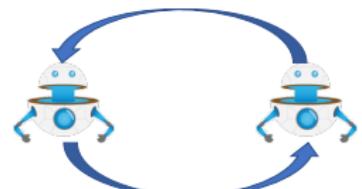
Conclusion

- Automated negotiation can enhance societal welfare.
- Genius and NegMAS as open-ended platforms for research in automated negotiation.
- Classical automated negotiation research in economics focused on simplified situations and provided performance guarantees.
- Many open questions:
 - General environment and unknown opponents.
 - Incomplete information about self.
 - Concurrent negotiations.
 - Negotiations with non-stationary utilities.
 - When to use negotiation?



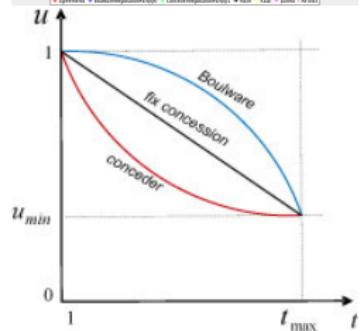
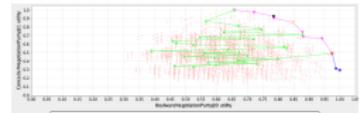
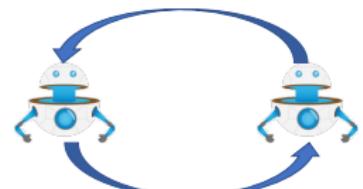
Conclusion

- Automated negotiation can enhance societal welfare.
- Genius and NegMAS as open-ended platforms for research in automated negotiation.
- Classical automated negotiation research in economics focused on simplified situations and provided performance guarantees.
- Many open questions:
 - General environment and unknown opponents.
 - Incomplete information about self.
 - Concurrent negotiations.
 - Negotiations with non-stationary utilities.
 - When to use negotiation?



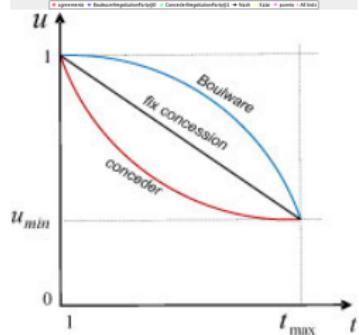
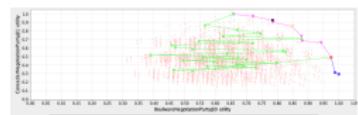
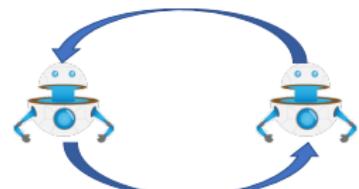
Conclusion

- Automated negotiation can enhance societal welfare.
- Genius and NegMAS as open-ended platforms for research in automated negotiation.
- Classical automated negotiation research in economics focused on simplified situations and provided performance guarantees.
- Many open questions:
 - General environment and unknown opponents.
 - Incomplete information about self.
 - Concurrent negotiations.
 - Negotiations with non-stationary utilities.
 - When to use negotiation?



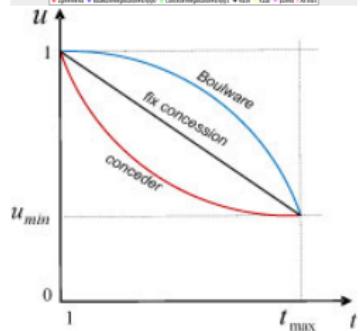
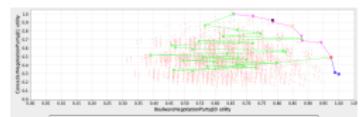
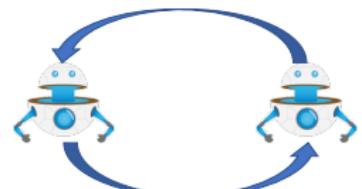
Conclusion

- Automated negotiation can enhance societal welfare.
- Genius and NegMAS as open-ended platforms for research in automated negotiation.
- Classical automated negotiation research in economics focused on simplified situations and provided performance guarantees.
- Many open questions:
 - General environment and unknown opponents.
 - Incomplete information about self.
 - Concurrent negotiations.
 - Negotiations with non-stationary utilities.
 - When to use negotiation?



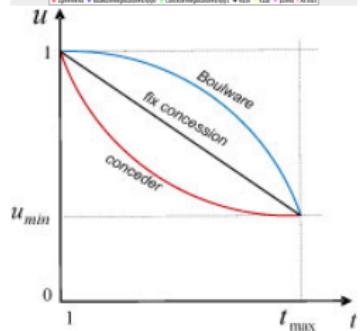
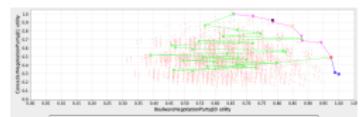
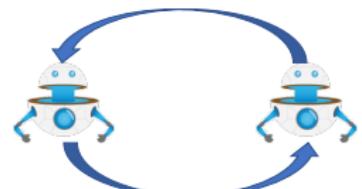
Conclusion

- Automated negotiation can enhance societal welfare.
- Genius and NegMAS as open-ended platforms for research in automated negotiation.
- Classical automated negotiation research in economics focused on simplified situations and provided performance guarantees.
- Many open questions:
 - General environment and unknown opponents.
 - Incomplete information about self.
 - Concurrent negotiations.
 - Negotiations with non-stationary utilities.
 - When to use negotiation?



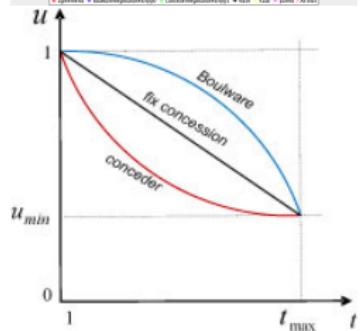
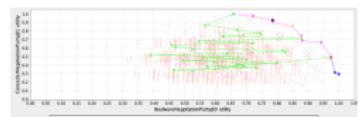
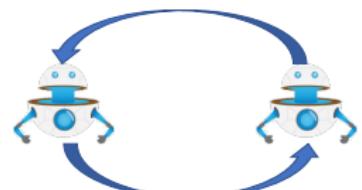
Conclusion

- Automated negotiation can enhance societal welfare.
- Genius and NegMAS as open-ended platforms for research in automated negotiation.
- Classical automated negotiation research in economics focused on simplified situations and provided performance guarantees.
- Many open questions:
 - General environment and unknown opponents.
 - Incomplete information about self.
 - Concurrent negotiations.
 - Negotiations with non-stationary utilities.
 - When to use negotiation?



Conclusion

- Automated negotiation can enhance societal welfare.
- Genius and NegMAS as open-ended platforms for research in automated negotiation.
- Classical automated negotiation research in economics focused on simplified situations and provided performance guarantees.
- Many open questions:
 - General environment and unknown opponents.
 - Incomplete information about self.
 - Concurrent negotiations.
 - Negotiations with non-stationary utilities.
 - When to use negotiation?



Thank you for listening (y.mohammad@aist.go.jp,
amy_greenwald@brown.edu)