

Development of an Automated Negotiation Agent for the ANAC 2026 SCML Standard Track

Amer Mohammad Nayef Dyab¹, Hamid Majidi Balanji²

¹Department of Electrical and Electronics Engineering, Özyeğin University

²Department of Computer Science, Özyeğin University

amer.thiab@ozu.edu.tr, hamid.majidi@ozu.edu.tr

Abstract

We present an automated negotiation agent for the SCML Standard sub-track of the ANAC 2026 competition. The agent manages production, procurement, and sales in the simulated SCML supply chain through four coordinated components: a time-aware Boulware concession strategy, a decoupled acceptance rule, a demand-driven production policy, and a partner classifier that assigns observed behavior to procurement archetypes drawn from real PCB sourcing practice. Controlled A/B experiments show that a small number of structural design choices drive almost all of the agent’s score, while several adaptive extensions added on top produced no measurable gain. Evaluated across a nine-seed production-scale tournament against the built-in SCML baselines, the agent achieves a mean net score of 1.054 ± 0.034 , ranking first against all baselines.

1 Introduction

Automated Negotiation, where software agents resolve conflicting demands through structured offer exchange [Jennings et al., 2001], is central to the ANAC competition, hosted at IJCAI 2026 in Bremen [Aydoğan et al., 2026]. The SCML Standard track places an agent as the manager of a single factory within a simulated supply chain [Mohammad et al., 2019]. On every simulated business day, the agent conducts concurrent bilateral negotiations over *price*, *quantity*, and *delivery day*, then translates the resulting contracts into a production plan [Mohammad et al., 2024a].

Unlike the OneShot sub-track [Mohammad et al., 2024b], the Standard track introduces temporal continuity: (i) contracts persist across days, (ii) inventory carries forward, and (iii) the same partners reappear repeatedly. This richer setting motivates the four-component design described here. The authors’ background in PCB procurement and electronics RFQs directly informs the partner-classification component, which is the principal novelty of this work.

2 Background

BOA framework. The dominant architecture for heuristic negotiation agents decomposes decision logic into **Bidding**,

Opponent modeling, and **Acceptance** [Baarslag et al., 2014]. Bidding strategies differ in their concession rate; Boulware-style strategies concede slowly early and steeply near the deadline [Faratin et al., 1998]. Recent work argues that bidding and acceptance should be governed by *separate* rules, since capturing value and securing agreements before the deadline are distinct objectives [Baarslag et al., 2014].

Opponent modeling. Approaches range from frequency-based estimators to full Bayesian formulations [Baarslag et al., 2016]. Rather than fitting a numerical utility function, the present agent classifies each partner into one of four behavioral archetypes, turning opponent modeling into an active strategy selector.

SCML environment. The framework is built on NegMAS [Mohammad et al., 2021] and exposes an Agent World Interface (AWI) that supplies daily targets, market prices, inventory state, and penalty rates. Negotiations follow the Alternating Offers Protocol (AOP) [Aydoğan et al., 2017]. The production strategy translates contracts into a daily manufacturing process. Simply put, the plan is the fourth BOA-plus component specific to the supply chain setting.

3 Agent Design

The agent separates negotiation logic from production logic, and within negotiation further separates bidding, acceptance, and opponent modeling. Every design decision was validated through controlled A/B experiments; a candidate change was retained only when it produced a measurable score improvement.

3.1 Bidding Strategy

The aspiration function follows the time-dependent Boulware model [Faratin et al., 1998]:

$$a(t) = 1 - t^{1/e_b}, \quad (1)$$

where $t \in [0, 1]$ is the elapsed fraction of the negotiation deadline and $e_b > 1$ is the bidding exponent (slow concession). The aspiration is mapped to a price by linear interpolation between the agent’s ideal price and a *market-anchored floor* p^* , set to the prevailing market trading price returned by the AWI. Anchoring at the market price rather than the issue’s extreme closes a structural value leak present in a textbook Boulware curve: even at the deadline, the agent never proposes below the market consensus.

3.2 Acceptance Strategy

The acceptance rule uses a *separate* exponent $e_a > e_b$, so the threshold descends faster than the proposal trajectory as the deadline approaches—encoding the asymmetry between value capture (bidding) and agreement securing (acceptance) [Baarslag et al., 2014]. An offer is accepted when two conditions hold simultaneously: (i) the offered quantity does not exceed the agent’s remaining daily requirement (over-commit guard), and (ii) the contract’s estimated utility u satisfies

$$u \geq a(t; e_a) \cdot u_{\max} \cdot \beta_{\text{arch}}, \quad (2)$$

where β_{arch} is a small archetype-dependent bias described in Section 3.4. The over-commit guard runs before utility reasoning: an oversized offer is rejected regardless of price.

Contract utility model. Utility nets revenue (or resale value on the buy side) against production cost and against shortfall or disposal penalties that over-commitment would trigger. Penalty rates are read from the AWI rather than assumed, so the model reflects the specific world instance. An offer can carry an attractive unit price and still be rejected if the quantity beyond the day’s requirement incurs a sufficient penalty to drive the net utility negative.

3.3 Production Policy

The production rule is demand-driven and conservative. On each simulated day the scheduled output is

$$q_{\text{prod}} = \min\left(q_{\text{unfulfilled}}^{\text{sell}}, q_{\text{inventory}}^{\text{input}}, q_{\text{capacity}}^{\text{line}}\right). \quad (3)$$

Nothing is manufactured without a committed downstream sale, eliminating disposal risk by construction. The decision to suppress speculative production entirely is motivated by empirical evidence (Section 5): speculative production was the principal driver of negative scores in early iterations.

3.4 Opponent Classifier

Skilled negotiators read the partner’s behavior in opening rounds before conceding [Baarslag et al., 2016]. The classifier implements this practice explicitly. As offers are exchanged, the agent accumulates three per-partner features:

- **Trust:** $1 - \text{CV}(\text{prices})$, measuring offer consistency (high = tightly clustered prices).
- **Price:** mean signed deviation of partner offers from the market price (positive = anchors above market).
- **Reliability:** fraction of concluded negotiations ending in agreement.

Once a minimum-observation threshold is met, these features are compared against fixed band thresholds to assign one of four archetypes:

Archetype	Signature	Bias β
Ideal Supplier	consistent, near market, reliable	< 1 (easier)
Budget Specialist	below market, consistent, reliable	1 (neutral)
Suspicious Cheap	below market, unreliable	> 1 (harder)
Premium	above market, consistent	1 (neutral)

Algorithm 1 Daily lifecycle of the implemented agent

- 1: Clear per-step caches; read AWI state
- 2: $q_{\text{prod}} \leftarrow \min(\text{unfulfilled sells, input inv., line cap.})$
- 3: Schedule q_{prod} units
- 4: **On propose:** compute t , evaluate $a(t; e_b)$, interpolate price, return offer
- 5: **On respond:** record offer in classifier buffer; apply over-commit guard; compute utility u ; evaluate threshold $a(t; e_a) \cdot u_{\max} \cdot \beta$; accept iff $u \geq \text{threshold}$
- 6: **On success/failure:** update sell history and per-partner outcome counts

Partners below the observation threshold are labeled *Unknown* and handled with the default posture. The bias β_{arch} raises or lowers the acceptance threshold by a small factor without overriding the economic judgment of the utility model.

4 Implementation

The agent is implemented as a single Python class inheriting from `StdAgent` (NegMAS 0.15.4, SCML 0.8.2, Python 3.11.9). The lifecycle has three layers: (i) one-time initialization (allocates per-partner observation buffers and sell-history records), (ii) a daily step hook (clears caches, reads AWI state, schedules production), (iii) and per-negotiation hooks (`propose` / `respond` / `success`–`failure` callbacks).

Algorithm 1 summarizes the daily control flow. The `propose` method evaluates Eq. (1) with e_b and interpolates to a price; the `respond` method applies the over-commit guard, then evaluates Eq. (2) with e_a . The classifier is a standalone function fed by offer buffers accumulated across callbacks, keeping it inspectable in isolation.

5 Experiments and Results

Evaluation setup. Two tournament scales are used:

- *Skeleton-scale* (3 seeds, 2 configurations/seed, 10 simulated days/run, 4 competitors) supports fast iteration.
- *Production-scale* (9 seeds, more configurations and runs per world) establishes the headline benchmark.

Opponents in both scales are the three built-in SCML baselines: (i) `GREEDYSTDAGENT` (heuristic, aggressive, strongest baseline), (ii) `SYNCRANDOMSTDAGENT` (random with cross-negotiation accounting), and (iii) `RANDOMSTDAGENT` (fully random, sanity check).

Score trajectory. Table 1 shows the accepted iteration sequence. The initial scaffold scored well below zero: speculative production triggered disposal penalties on every run. Switching to demand-driven production was necessary but not sufficient; the negotiation layer was rejecting most offers. *Market-anchored pricing*—moving the Boulware asymptote from the issue extreme to the AWI market price was the decisive change, lifting the mean score above 1.0 for the first time. The Boulware concession curve added a consistent further gain, decoupling the acceptance exponent was retained on

Table 1: Score trajectory of accepted iterations (skeleton-scale, 3 seeds).

Iteration	Mean	Std
Initial scaffold	-0.189	—
Demand-driven production	-0.217	—
Market-anchored pricing	1.011	0.050
Boulware propose	1.035	0.069
Decoupled respond	1.034	0.076

Table 2: Score of rejected experiments (skeleton-scale, 3 seeds).

Experiment	Mean	Std
Buy/sell coupling	0.828	0.057
Coupled respond ($e_a = e_b$)	1.011	0.047

methodological grounds (encoding the bidding–acceptance asymmetry) with score preserved within noise.

Table 2 records two experiments that were ultimately rejected. Buy/sell coupling, gating buy-side decisions on internally tracked sell contracts rather than the AWI signal, regressed the score to 0.828 due to midday inconsistency between the internal state and the AWI. Using a symmetric exponent ($e_a = e_b$) held the acceptance threshold near the ideal price for most of the negotiation, timing out otherwise acceptable offers; the result was indistinguishable from the locked baseline and was reverted.

Final leaderboard. Table 3 reports the skeleton-scale ranking. The agent leads all built-in baselines by a 22-percentage-point margin over GREEDYSTDAGENT. The production-scale sweep over nine seeds confirms robustness: mean score 1.054 ± 0.034 .

6 Discussion

Four design lessons emerge from the experiment record. First, speculative production was the dominant source of negative scores; Demand-driven scheduling was the precondition for any negotiation-side improvement to become visible. Second, the AWI is the authoritative source of daily targets; any internal accounting that duplicates it introduces race conditions, as the buy/sell coupling regression illustrates. Third, the placement of the concession asymptote is decisive: anchoring at the issue extreme silently leaks value at the deadline, and moving it to the market price closed that leak in a single change. Fourth, identical bidding and acceptance exponents hold the threshold too high for too long; decoupling them allows ambitious proposing alongside pragmatic accepting.

Two caveats apply. The built-in SCML baselines are static or random; no adaptive behavioral pattern exists for the classifier to exploit, which is why the archetype component cannot demonstrate a score gain in the current harness and is retained on the strength of its design. The null results on adaptive layers should therefore be read as evidence about the *test environment* as much as the features themselves. Validation against adaptive opponents, including entries from past ANAC competitions, is the clearest next step.

Table 3: Leaderboard, skeleton-scale tournament.

Agent	Mean	Std	Rank
Our Agent	1.034	0.076	1
SYNCRANDOMSTDAGENT	0.913	0.096	2
GREEDYSTDAGENT	0.849	0.155	3
RANDOMSTDAGENT	-0.011	0.242	4

7 Conclusion

We presented a four-component negotiation agent for the ANAC 2026 SCML Standard track, combining a Boulware bidding strategy, a decoupled acceptance rule, a demand-driven production policy, and a partner classifier grounded in real PCB sourcing archetypes. Across a nine-seed production-scale tournament the agent scores 1.054 ± 0.034 , ranking first against all built-in SCML baselines. The principal novelty is the opponent classifier, which reframes opponent modeling as an active strategy selector rather than a passive utility estimator; its value against adaptive ANAC opponents remains to be demonstrated. The experiment record itself, documenting both accepted changes and informative null results, constitutes a secondary contribution as a design leverage map for future SCML agents.

Acknowledgments

The authors thank the course instructor of CS 551 (Introduction to Artificial Intelligence, Özyeğin University, Spring 2026) for the discussion that motivated the migration from the OneShot to the Standard sub-track.

References

References

- [Aydoğan et al., 2017] Reyhan Aydoğan, David Festen, Koen V. Hindriks, and Catholijn M. Jonker. Alternating offers protocols for multilateral negotiation. In *Modern Approaches to Agent-based Complex Automated Negotiation*, volume 674 of *Studies in Computational Intelligence*, pages 153–167. Springer, Cham, 2017.
- [Baarslag et al., 2014] Tim Baarslag, Koen Hindriks, Mark Hendriks, Alexander Dirkzwager, and Catholijn M. Jonker. Decoupling negotiating agents to explore the space of negotiation strategies. In *Novel Insights in Agent-based Complex Automated Negotiation*, volume 535 of *Studies in Computational Intelligence*, pages 61–83. Springer, Tokyo, 2014.
- [Baarslag et al., 2016] Tim Baarslag, Mark J. Hendriks, Koen V. Hindriks, and Catholijn M. Jonker. Learning about the opponent in automated bilateral negotiation: A comprehensive survey of opponent modeling techniques. *Autonomous Agents and Multi-Agent Systems*, 30(5):849–898, 2016.
- [Faratin et al., 1998] Peyman Faratin, Carles Sierra, and Nicholas R. Jennings. Negotiation decision functions for autonomous agents. *Robotics and Autonomous Systems*, 24(3–4):159–182, 1998.

- [Jennings et al., 2001] Nicholas R. Jennings, Peyman Faratin, Antonis R. Lomuscio, Simon Parsons, Michael Wooldridge, and Carles Sierra. Automated negotiation: Prospects, methods and challenges. *Group Decision and Negotiation*, 10(2):199–215, 2001.
- [Mohammad et al., 2019] Yasser Mohammad, Enrique Areyan Viqueira, Nahum A. Ayerza, Amy Greenwald, Shinji Nakadai, and Soichiro Morinaga. Supply chain management world. In *PRIMA 2019: Principles and Practice of Multi-Agent Systems*, volume 11873 of *LNCS*, pages 153–169. Springer, Cham, 2019.
- [Mohammad et al., 2021] Yasser Mohammad, Shinji Nakadai, and Amy Greenwald. NegMAS: A platform for automated negotiations. In *PRIMA 2020: Principles and Practice of Multi-Agent Systems*, volume 12568 of *LNCS*, pages 343–351. Springer, Cham, 2021.
- [Mohammad et al., 2024a] Yasser Mohammad, Amy Greenwald, and Shinji Nakadai. Supply chain management league (standard). ANAC 2024 SCML competition specification, 2024. <https://scml.cs.brown.edu/files/scml/y2024/scml2024.pdf>.
- [Mohammad et al., 2024b] Yasser Mohammad, Amy Greenwald, and Shinji Nakadai. Supply chain management league (oneshot). ANAC 2024 SCML competition specification, 2024. <https://scml.cs.brown.edu/files/scml/y2024/scml2024oneshot.pdf>.
- [Aydoğan et al., 2026] Reyhan Aydoğan, Tim Baarslag, Katsuhide Fujita, and Catholijn M. Jonker. ANAC 2026: 17th automated negotiating agents competition. IJCAI 2026, Bremen, Germany, August 15–21, 2026. <https://web.tuat.ac.jp/~katfuji/ANAC2026/>.