PIRASA: Strategic Protocol Selection for E-Commerce Agents

Jack Hopkins[1], Özgür Kafalı[2], Bedour Alrayes[3], and Kostas Stathis[4]

[1] University of Cambridge, Advanced Computer Science
The Old Schools, Trinity Ln, Cambridge CB2 1TN, UK
jack.hopkins@me.com

[2] School of Computing, University of Kent
Canterbury, CT2 7NF, UK
R.O.Kafali@kent.ac.uk

[3] King Saud University, Information Technology
King Khalid Rd, Riyadh, Saudi Arabia
balrayes@ksu.edu.sa

[4] Royal Holloway, University of London, Computer Science
Egham, TW20 0EX, UK
kostas.stathis@rhul.ac.uk

Abstract

We present PIRASA: an agent-based simulation environment for studying how autonomous agents can best interact with each other to exchange goods in e-commerce marketplaces. A marketplace in PIRASA enables agents to enact buyer or seller roles and select from sales, auction, and negotiation protocols to achieve the individual goals of their users. An agent's strategy to maximize its utility in the marketplace is guided by its user's preferences and constraints such as 'maximum price' and 'deadline', as well as an agent's personality attributes, e.g., how 'eager' or 'late' the agent can be for exchanging goods and whether the agent is a 'spender' or 'saver' in an exchange. To guide the agent's actions selected by a strategy, we use the notion of electronic contracts formulated as regulatory norms. In this context, we present how PIRASA is organized with regards to seller processes for goods submission, the inclusion of buyer preferences, and the management of transactions through specialized broker agents. Using randomized simulations, we demonstrate how a buyer agent can strategically select the most suitable protocol to satisfy its user's preferences, goals and constraints in dynamically changing market settings. The generated simulation data can be leveraged by researchers to analyze agent behaviors, and develop additional strategies.

Keywords: Agent-based e-commerce; Protocol selection; Electronic contracts; Simulation.

1 Introduction

The evolution of the Internet has brought about new standard procedures for purchasing goods online using Web browsers or mobile applications (Alt & Zimmermann, 2016; Spiekermann, Böhme, Acquisti, & Hui, 2015). Using these procedures, referred to commonly as protocols, users can bid on items they would like to buy in auctions, they can negotiate with the sellers directly to get the best possible price, and they can order at a fixed price to get the item quickly. Since there are many alternative forms of such protocols to choose from, it is important that users can strategically select the best one for their needs. Consider the following scenario:

Example 1. *Bob wants to buy a present for his father as father's day is approaching. He has a watch in mind that he thinks his father would like. He goes online, searches for potential sellers and finds two: One has the watch in stock and provides a direct sales protocol for purchasing the watch at the standard price for next-day delivery, whereas the other provides an auction protocol which allows for buying the watch at a much lower price but it could potentially take much longer time to complete. Given the time constraint Bob has due to father's day, he decides to purchase the watch from the first seller, using the* direct sales *protocol.*

It is becoming increasingly common for virtual assistant systems such as 'Amazon Echo' and 'Google Home' to contain basic integration into e-commerce platforms in addition to their information retrieval capabilities. Although these systems have the capability to purchase items for a fixed price, they are unable to enact strategies for the acquisition of goods in e-commerce markets with varying protocols (e.g., Auctions). With recent advances in agent technology and machine learning, such *personal agents* should be able to make decisions on behalf of their users' goals, preferences and constraints. An agent should be able to search through available marketplaces for potential sellers, and determine which protocol would result in a purchase that meets the user's needs, possibly based on further interactions with other agents, whether human or artificial. The issue then becomes, if we had to build such an agent, how to determine in an electronic market which protocol is most suitable to best satisfy the specific user's needs.

We present PIRASA: An agent-based simulation framework that is designed to test different strategies in dynamic electronic marketplaces, and evaluate which protocol is the most preferable given the user's goals, preferences, and constraints. For example, an agent can be designed to help Bob choose the best protocol for purchasing a watch for his father. Such an agent should understand that Bob's time constraint is more important (since father's day is approaching) than the price of the watch. Therefore, the agent should decide to purchase the watch from the seller offering a direct sales protocol. However, if the watch offered by the direct sales protocol is significantly more expensive than purchasing the watch through an auction, then Bob's agent might reconsider its decision. Agents in PIRASA additionally have personality traits. For example, some buyers are eager to buy an item, whereas others prefer to save money when purchasing items online. Similarly, some sellers offer fast delivery times, whereas others offer good deals via flexible pricing options. Each seller also has a preferred sales type chosen among three general purchase protocols: direct sales, negotiation, and auction. PIRASA supports endless protocol configurations based on the protocol attributes, e.g., duration which determines the amount of time before the protocol terminates.

We use the notion of electronic contracts to regulate the interactions between the agents in a protocol, which are formalized via social norms represented as commitments, authorizations, and prohibitions (Singh, 2013). Agent reasoning about norms is provided via the Event Calculus (Chesani, Mello, Montali, & Torroni, 2013; Kafalı & Yolum, 2016). Following Example 1, Bob can

check the delivery status of the watch via his contract with the seller. If something goes wrong and delivery is missed for father's day, the seller should commit to issuing a refund for Bob. A normative approach provides a high level of organizational flexibility, where the involved parties can create or cancel a norm, release it, or delegate it to others.

We propose to perform strategic protocol selection via simulations. Once presented with a set of alternative purchase protocols and market options, the buyer agent simulates its potential interactions in each available protocol and records its utility, which results in a ranking of the protocols. The multiagent simulation infrastructure of our framework relies upon the JADE agent platform (Bellifemine, Poggi, Rimassa, & Turci, 2000), on top of which we have provided a user interface to configure various properties of the simulation. PIRASA incorporates dynamic market elements and agent autonomy into the simulation environment. Design-time solutions such as model checking (Bataineh, Bentahar, Menshawy, & Dssouli, 2017; Montali, Calvanese, & De Giacomo, 2014) fail to incorporate such autonomy, and can only verify whether an agent's goals comply with the protocol specification. To the best of our knowledge, this paper is the first attempt to tackle this problem from a run-time perspective.

Our contributions are as follows:

- We provide a simulation framework for agent-based electronic commerce, where buyer agents interact with seller agents through broker agents. The broker agents regulate the interactions between the agents (formalized via regulatory norms) in the market.
- We generate multiple seller behaviors using the available agent traits in the framework for three protocol types; direct sales, negotiation and auction.
- We create multiple market settings, which reflect different levels of competitiveness and purchase options for the buyers, and simulate the buyer agents' strategies in those settings. We propose hypotheses on how buyer agents should behave in the presence of other buyers, and verify the hypotheses using buyer utilities gathered from the simulation data.
- Other researchers can benefit from the generated simulation data to analyze buyer and seller behaviors, and develop additional strategies.

2 PIRASA Structure and Overview

Parallel to the growing of electronic markets, practical e-commerce systems *need* to adapt to end users' preferences. To facilitate their construction, it is first necessary to develop a framework for the automated simulation, evaluation and comparison of strategies for the purchase of goods in markets with varying protocols.

The *problem domain* discussed in this work is e-commerce agent-based market simulation, whose *solutions* are instances of *practice-inspired research* (Sein, Henfridsson, Purao, Rossi, & Lindgren, 2011). The practice that motivates our work is *online shopping*. In current state of the art, a human user visits a number of websites to choose the best buying option based on her needs. For example, if she wants to buy a watch, she can order it from "Amazon Prime" with "Next Day Delivery" option, or buy the same watch from an "eBay" auction for a lower price but with longer order processing time (e.g. wait for the auction to finish). Reasoning about the tradeoffs among such choices results in a loss of time for the user. Our proposal acts as a *recommender system* for automating this

tedious process by using intelligent agents that know the preferences of their users. Specifically, we propose a framework, PIRASA, to support negotiation, auction, and purchasing protocols, and the capacity of agents to conduct strategies for optimising their users' market utilities.

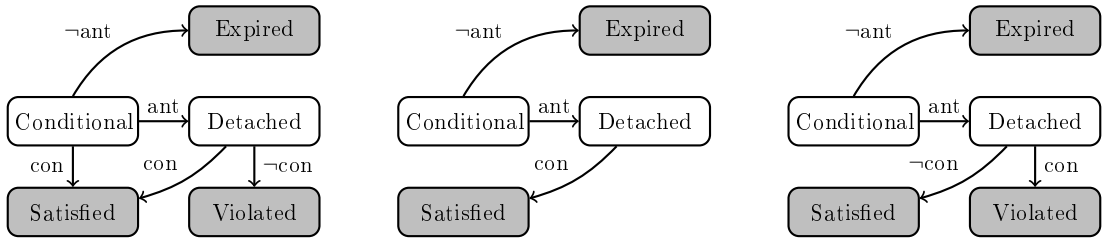
To determine the necessary attributes of a system to solve this problem, we aimed to develop a system that could simulate the common protocols found in modern e-commerce markets with sufficient agents to simulate protocols in even the most crowded of markets (Akula & Menascé, 2004). In order to motivate the agent's strategy, it is also necessary to imbue the agent with two key constraints that humans experience in traditional markets, i.e. time and cost.

Section 3 reviews the technical background for PIRASA, and Section 4 describes the various components of its implementation. Section 5 demonstrates our practical findings through simulation experiments, and discusses the potential integration of real-world e-commerce datasets to infer simulation parameters. Table 1 summarises the application stages of our use case, inspired by Action Design Research (ADR) methodology (Sein et al., 2011). We describe in detail the prototype implementation and evaluation stages of PIRASA in Sections 4 and 5; respectively.

Stage	**Description**	**Artifact**
Stage 1: Problem Formulation		
Practice-Inspired Research	Driven by the need for automation in online shopping decision-making process.	Shortcomings for automation and tool support for human users.
Theory-Ingrained Artifact	Artificial intelligence and agent-based models.	Agent development frameworks.
Stage 2: Prototype Development		
Market design	Literature survey (Alrayes, Kafalı, & Stathis, 2017) to gather parameters for realistic electronic markets.	Conceptual design of the online shopping market.
Evaluation	Agent-based simulation environment to evaluate the accuracy of protocol choices.	Alpha Version: Prototype implementation in JADE.
Stage 3: Reflection and Learning		
Analysis of Results	Recognition of limitations regarding the current prototype. Involve human users in the next phase. Interview online shoppers to better understand needs.	Beta Version: Improve prototype to include user-friendly interface and human-controlled agents.

Table 1: ADR methodology for online shopping use case.

When comparing with existing work, a number of different criteria can be used, namely: (i)



(a) Commitment (b) Authorization (c) Prohibition

Figure 1: Lifecycle of norms. Dark rectangles represent terminal states, i.e., the norm's lifecycle ends in those states.

The capacity of the system to operate with large numbers of agents; (ii) the number of e-commerce protocols that can be simulated; (iii) the flexibility of agents to mirror the myriad constraints and preferences that influence human behaviour when operating in a market; and (iv) the ease with which new agent strategies can be constructed. PIRASA is designed with these criteria in mind, and is compared in Section 6 with the relevant literature on simulation environments and e-commerce platforms.

3 Technical Background

In this section, we review the technical background that our simulation framework is built upon; electronic contracts and their formalization as regulatory norms, agent reasoning based in the Event Calculus, and the agent development platform JADE.

3.1 Electronic Contracts and Norms

A contract describes how the involved parties should act in a business dealing. We adopt social norms (Singh, 2013) to formally represent electronic contracts. Norms (commitments, authorizations, and prohibitions) take their basis from deontic logic concepts (Von Wright, 1999), and have been widely used in fields of artificial intelligence that deal with legal concepts (Boella & van der Torre, 2008; Dechesne, di Tosto, Dignum, & Dignum, 2013), compliance checking (Governatori, 2013), and requirements engineering (Kafalı, Ajmeri, & Singh, 2016).

Formally, a norm n(X, Y, antecedent, consequent) represents a social relationship between its subject (X) and object (Y) regarding its consequent when its antecedent holds. Here, n is the norm type (c for commitment, a for authorization, or p for prohibition), X and Y are agents, and the antecedent and the consequent are first-order logic predicates (either atomic propositions, or conjunctions or disjunctions of them). We model *conditional*, *detached*, *satisfied*, and *violated* norm states. Figure 1 describes the lifecycle of norms (Kafalı, Singh, & Williams, 2016). A conditional norm is detached when its antecedent holds. Satisfaction and violation conditions are described according to the norm type.

In this paper, we mainly adopt the commitment norm as the basis for representing electronic contracts. Commitments have previously been used in e-commerce (Kafalı & Torroni, 2012; Kafalı & Yolum, 2016). Consider the following commitment:

$$c(store, customer, payment, delivery) \tag{1}$$

The above commitment is a conditional commitment; if the antecedent (payment) is satisfied, then the subject (store) becomes committed to the object (customer) for satisfying the consequent (delivery), and the commitment becomes detached. A base-level commitment is simply a commitment with its antecedent condition being true. If the consequent is satisfied, the commitment is satisfied. After the commitment is detached, if the consequent is not satisfied, the commitment is violated.

3.2 Event Calculus

Event Calculus (EC) (Kowalski & Sergot, 1986) is an extension of first-order logic to interpret and reason about events in time. Table 2 summarizes the domain-independent axioms of EC. Predicate happens records events with the time points of their occurrence. Predicate initially specifies fluents that hold initially. Predicate holds_at queries the happened events to check whether a fluent holds at a specified time point. Predicate initiates marks that an event initiates a fluent at a specified time point. Predicate broken checks whether a fluent is terminated during a time period. Predicate terminates marks that an event terminates a fluent at a specified time point.

Table 2: Domain-independent axioms of the Event Calculus.

Predicate	**Description**
happens(E, T)	Event E happens at Time T
initially(F)	Fluent F is true at Time 0
holds_at(F, T)	Fluent F is true at Time T
initiates_at(E, F, T)	Event E initiates fluent F at Time T
broken(F, Ts, Te)	Fluent F is made false between times Ts and Te
terminates_at(E, F, T)	Event E terminates fluent F at Time T

We adopt the Reactive Event Calculus ($\mathcal{REC}$) (Chesani et al., 2013) as a logic programming tool that extends EC for run-time monitoring. The $\mathcal{REC}$ engine takes as input (i) a *normative theory* shared amongst all agents that describes how norms change state; (ii) a *protocol description* specific to each individual agent that describes the domain, e.g., consequences of the agents' actions as well as any known facts; and (iii) a *narrative* specific to each individual agent that contains the events performed through the evolution of time.

Listing 1 demonstrates a sample narrative in EC. According to the recorded events, the customer has paid for the item at time 4, the store has processed the order at time 5, and the courier has delivered the item at time 7. Like protocol descriptions, event traces are agent-dependent. That is, each agent is aware of only the events that are relevant, but does not see the events that might have happened for other agents.

Listing 1: Sample narrative in EC.

```
happens(pay(customer, store, item), 4).
happens(process_order(store, item), 5).
happens(deliver(courier, customer, item), 7).
```

Once the $\mathcal{REC}$ engine is run with above input, it produces an outcome that demonstrates the fluents the agent is aware of through time (e.g., states of commitments). $\mathcal{REC}$ can be extended with additional functionality besides commitment tracking such as exception handling behavior (Kafalı & Torroni, 2012; Kafalı, 2012).

3.3 JADE Platform

Agent-Based simulation is a widely adopted technique in distributed artificial intelligence to analyze agent behaviors and strategies. Amongst many agent development and simulation environments, we adopt JADE (Bellifemine et al., 2000) to develop our agents as it provides reliable agent communication and documentation support. Moreover, despite the existence of a number of other Java agent development platforms (Luke, Cioffi-Revilla, Panait, & Sullivan, 2004; Baumer, Breugst, Choy, & Magedanz, 1999; Xu & Shatz, 2003), JADE is actively maintained and compliant to the FIPA[1] agent standard. JADE provides a library of Java classes to develop agent strategies as well as graphical interfaces to configure and run simulations. JADE supports asynchronous messaging for agent communication, and provides yellow pages for publish & subscribe type services to simulate electronic markets.

4 Pirasa Framework

E-commerce transactions in the real world such as listing and purchasing of items are simplified for buyers and sellers on sites such as Amazon and eBay by acting as a hub between the agents. We propose to mimic such transactions on the web by adding "broker" agents in addition to the sheer number of potential buyers and sellers. The inclusion of brokers simplifies the process of a potential buyer finding a seller, and therefore both increases the throughput of the system, while minimizing the amount of time an agent is actively looking for a new transaction. This architecture is reflected in Figure 2.

4.1 Agents and Transactions

Pirasa supports three types of agents: sellers, buyers, and brokers.

(i) Sellers and product submission: At the beginning of a simulation, broker agents solicit the seller agents to publish all items they want to sell through the broker in the form of *services*. In order for a seller to submit their item for sale to a broker, the seller first locates all brokers and then sends each one a message asking them to host the item. This seller protocol is shown in Figure 3. Each broker replies either rejecting the proposal, or returning a potential protocol with which to host the item. The seller can then select the protocol which best suits its goal, e.g., whether they want to maximize profit or prefer a quick transaction. At this point, a message of confirmation containing details about the item is sent to the preferred broker. The broker then replies with a confirmation

[1] http://www.fipa.org/

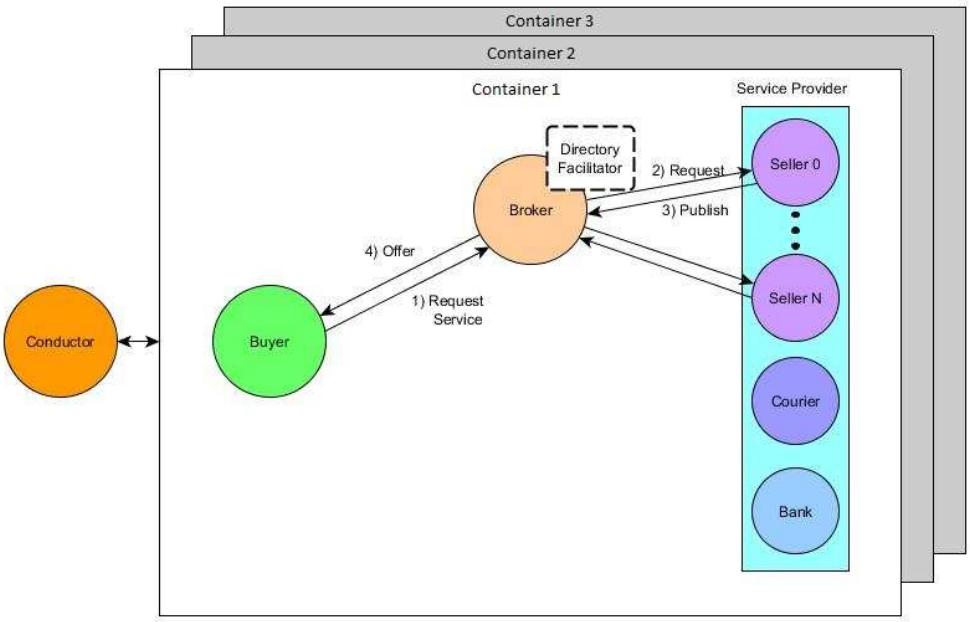


Figure 2: A service oriented architecture showing the buyer engaging with sellers to find a service.

of receipt, and instantiates an unactivated *service*. Potential buyers can now attempt to obtain the item, which triggers the activation of the service, starting the count down to its termination.

(ii) Buyers and user preferences: The formulation of buyer agents' goals enables constraints to be placed on how a buyer goes about obtaining items. A goal can currently be constrained by a *time limit*, a *price limit*, or both. For example, if an agent has a time constraint (i.e., deadline) for an item and there is an auction which will only end after the time constraint has been violated, then the buyer will ignore it. However, a buyer will not make any assumptions with regards to the total duration of a negotiation, as it is possible at any point for the seller to accept an offer, and end the negotiation. PIRASA currently supports the following four attributes for buyers:

- *Eagerness:* The propensity of an agent to place a high urgency on a transaction. A high eagerness modifier results in agents aiming for the quickest transactions possible.
- *Lateness:* The propensity of an agent to place a low urgency on a transaction.
- *Spending:* The propensity for an agent to enter into monetarily unfavorable transactions in exchange for eagerness.
- *Saving:* The propensity for an agent to enter only into highly favorable transactions.

(iii) Brokers and market transactions: Buyers initiate market interactions by requesting all broker agents for services which match their goals (see Figure 2). Upon perceiving the services available, a buyer strategically determines the best service to subscribe to, in accordance with its personality traits. Depending on the nature of the service, upon the receipt of a new offer from a buyer, the broker enforces the rules of the protocol in terms of a set of norms. If accepted, the

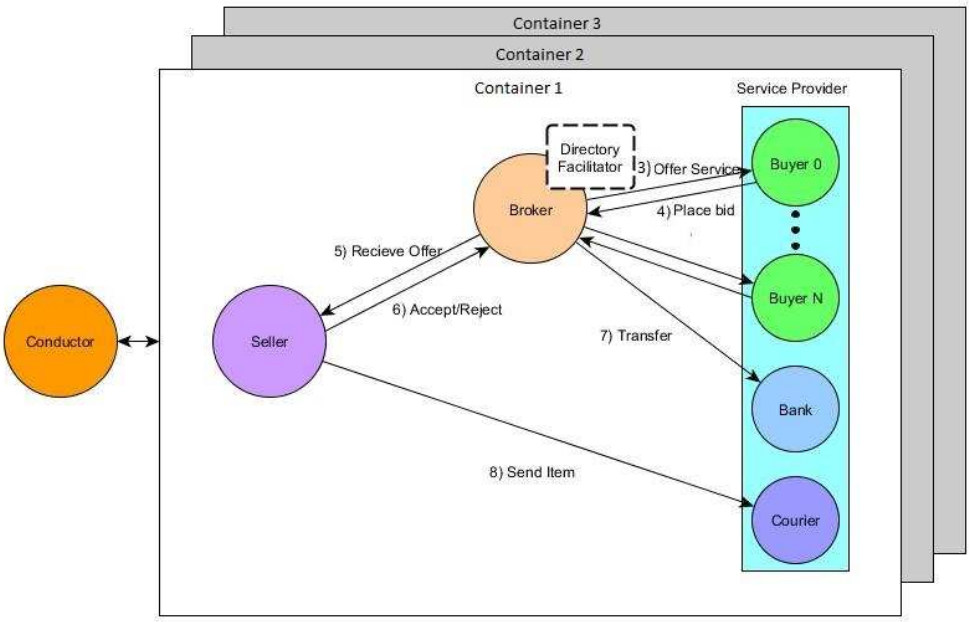


Figure 3: Negotiation protocol: Seller agent handling offers placed by buyer agents.

buyer engages in a contract with the seller to receive the item in the allotted time. Buyer agents are affected by two factors when reasoning on a service. Their personality traits either increase or decrease their tolerance of higher prices or service durations. Their goals enforce constraints over the cost and duration of possible services.

4.2 Predefined Protocols

A *protocol* in PIRASA is defined as a seven-tuple containing two integers and five boolean variables (described in detail in Section 4.3), which represent fundamental parameters of real world transaction protocols. The protocol description acts as a blueprint for the creation of a service, which in turn is the construct that actually handles offers from the buyer agents, and establishes norms among the buyer and seller agents. PIRASA supports three predefined protocols.

(i) Direct sales: A *direct sales* protocol is the simplest common method of transaction. A buyer offers the broker the asking price for the item and the protocol completes immediately. The protocol results in an item being sold for the market price with little time spent (not taking into account delivery times). Direct sales is favored by agents that have personalities with a high "eagerness" value. The below conditional commitment is created between the buyer and seller.

$$c(seller, buyer, payment, delivery) \tag{2}$$

When the seller is notified of payment, it is committed to ensuring the delivery of the item. The progression of commitment states is handled by each agent's $\mathcal{REC}$ engine. Note that the commitment theory is not the focus of this paper. However, we still record the number of commitment violations

per agent as part of our simulation data.

(ii) Auction: An *auction* is defined as an interaction between any number of buyers and a single seller that lasts for a predetermined time, mediated by a broker. Technically, the auction is regarded as a single-item, first-price, open-cry, ascending auction (Parsons, Rodriguez-Aguilar, & Klein, 2011) (Harris & Raviv, 1981). An auction is started as soon as the seller accepts the proposal from the broker to host it, and during its lifecycle the broker receives bids from any buyer agent. The broker does not interact with the seller during this time, and therefore can accept or reject an offer based on whether or not the offering agent has violated any norms. Once a buyer has its offer accepted by the broker, the following norms are created amongst the buyer, seller, and broker.

$$a(buyer, broker, true, bid) \tag{3}$$
$$c(buyer, seller, highest\_bid, payment) \tag{4}$$
$$c(seller, buyer, payment, delivery) \tag{5}$$

Norm 3 states that all buyers are authorized to make bids on the auctioned item. Norm 4 states that the buyer with the highest bid is committed to sending the payment to the seller. This commitment ensures that there is no way for a buyer to retract a bid (that has not been outbid) without violating their commitment. Norm 5 is the same commitment from the direct sales protocol (Norm 2) that handles delivery of the item once it is paid for.

(iii) Negotiation: A *negotiation* protocol is somewhat similar to the auction protocol outlined above, in that it is better for the buyer when it is not competing with others for an item, otherwise it is advantageous for the seller. The negotiation architecture is demonstrated in Figure 3. Since the seller has control over whether or not it will accept an offer, it can make counter-offers to buyers which will result in the item being sold below its market value, or alternatively the seller can wait for a long time to receive a higher offer by rejecting all lower offers. Certainly, this is a riskier strategy, because if a seller overvalues their item, it could result in no sale being made at all. In PIRASA, the propensity for sellers to adopt such strategies is based on their personalities, which are determined by attributes such as greed and eagerness. The following norms are relevant for the negotiation protocol.

$$a(buyer, seller, reject\_offer, offer) \tag{6}$$
$$c(buyer, seller, accept\_offer, payment) \tag{7}$$
$$c(seller, buyer, payment, delivery) \tag{8}$$

Norm 6 states that the buyer is authorized to make an offer if its previous offer is rejected by the seller. Norm 7 states that the buyer is committed to sending the payment to the seller if the offer is accepted. Norm 8 is the same commitment from the direct sales protocol (Norm 2) that handles delivery of the item once it is paid for.

4.3 Custom Protocols

It is crucial for market designers to customize the protocols available in their market place to attract a variety of buyers and sellers. In addition to the predefined protocols, PIRASA supports the creation

of new protocols via customization of protocol attributes. The attributes of a service encapsulating a protocol are the following:

- Max clients: The maximum number of buyers which can subscribe to the service. In the case of an an auction, for example, there would be no limit, but for a sale, the maximum number would be one.
- Max length: The maximum number of time steps that the service lasts. In the case of an auction, it would be a finite number, whereas for a negotiation it would be infinite, as a negotiation lasts as long as the two parties want it to before the transaction is finalized.
- Seller involvement: Whether the broker is authorized to make decisions on the seller's behalf. For example, if during a negotiation the broker can accept an offer made by the buyer, or whether they have to forward the message to the seller for approval.
- Alternating offers: Whether the seller is authorized to make counter offers to the buyer. Used exclusively with seller involvement.
- Buyer informed: Whether the broker informs buyers when they are outbid.
- Activate immediately: Whether a service is activated immediately, or whether it is activated when a bid has first been placed on it.
- One offer per buyer: Whether a buyer is authorized to have more that one outstanding offer on a service.

These attributes enable a variety of protocols to be formulated, including direct sales, auctions, and negotiations. When looking for a particular item, buyers look for a protocol for that specific item. They can either activate a dormant service, or join a running one (assuming that the max-clients variable is greater than than the number of agents the have already placed offers). A protocol is initially created dormant, with no buyers subscribed to it. As soon as the first buyer subscribes, the protocol is activated and it exists only for as long as its max length is not surpassed. A protocol can also be automatically started when the broker starts to host it.

5 Experiments

5.1 Experimental Design

To highlight the use of PIRASA, we construct a set of experiments to test the individual utility of several buyer agents. Each experiment is repeated 100 times with the average utility being measured and compared amongst buyers. The utility metric identifies the effectiveness of buyer agents, which corresponds to the goal completion rate of buyers. Naturally, this metric is most useful for simulations in which there are fewer number of sellers.

Buyers: We describe three buyer personalities with the following parameters:

- EagerSaver: agents have eager and saver traits of 0.7.
- EagerSpender: agents have eager and spend traits of 0.9 and 0.7, respectively.

- Saver: agents have eager and saver traits of 0.4 and 0.9, respectively.

Sellers & Protocols: In our experiments sellers are described by the protocols that they support in order to sell their goods.

- *Sales* – allows sellers to sell goods at a fixed price;
- *Auction* – lets the market of buyers determine the price of goods; and
- *Negotiation* – supports sellers to negotiate with buyers the price of goods.

In Table 3 we show how the protocols compare with one another.

Table 3: Comparison of the Seller protocols we consider.

	Sales	Auction	Negotiation
Max clients	1	20	∞
Max length	0	∞	∞
Seller involvement	✗	✗	✓
Alternating offers	✗	✗	✓
Buyer informed	✗	✓	✓
Activate immediately	✗	✓	✗
One offer per buyer	✗	✗	✓

In addition to the creation of the above scenarios, we capture data from human users in historical Ebay auctions (Jank & Shmueli, 2010, 2017), specifically the auctions of Cartier watches that lasted 7 days. We transform this data to agent traits and constraints, enabling the simulation and permutation of historical auctions, for the purpose of evaluating whether historical outcomes could be improved. From historical bids, we normalise the bid price and bid time, deriving the proportion of an auction that has elapsed, and the proportion of the end price that has been achieved at every bid. We then regress bid time against bid price for each bidder, and capture the resultant parameters as 'Eager' and 'Spender/Saver' traits for each participant in the auction, applying 'Max Price' constraints derived from the maximum bid that each user made. The above development is reflected in the following subsections, where we describe our simulations and results.

We applied simulated annealing to derive the agent attributes, implemented using the Opt4j library (Lukasiewycz, Glaß, Reimann, & Teich, 2011). Our goal is to determine the optimal parameters for each agent in a PIRASA simulation, such that the output bid ordering is the same as an equivalent historical auction - thus demonstrating that PIRASA is able to simulate the dynamics of historical auctions, with few parameters. Formally, given a simulation with n agents ($n \geq 2$), our aim is to minimise $f'(o, h)$, the mean-square of the differences of indices of each element between o and h, where o is the output of a PIRASA run; a sequence of bids $\mathbf{o} = \{o_1, o_2, ..., o_n\}$, and h is the sequences of bids from a historical auction or negotiation $\mathbf{h} = \{h_1, h_2, ..., h_n\}$. We randomly initialise a set of parameter vectors, representing Eagerness ($\mathbf{e}$), Lateness ($\mathbf{l}$), Spender ($\mathbf{sp}$) and Saver ($\mathbf{sa}$). We sample the output from a PIRASA run, determined by these parameters, and generate a fitness value from the mean-square differences, with which we can continue annealing.

For each of the three experiments, we reran the annealing to estimate the optimal parameters for that scenario. As we are not attempting to learn a generalisable function of PIRASA, but rather learn the best parameters for a given auction, over-fitting did not present an issue.

5.2 Simulations

For all experiments, we simulate the selected buyers in all three protocols and report the average utilities. Moreover, we state our hypotheses informally, and investigate whether they hold.

Simulation 1: We examine a buyer agent with an "EagerSaver" personality. There are no other buyer agents (as competitors) in this setting.

Hypothesis 1: We expect the agent to prefer faster (due to eager) and cheaper (due to saver) services. Therefore, we anticipate that the agent would gain higher utility in either direct sales or negotiation protocols. Since there is no competition from other buyers, the simulations should result in a low mean spend for the agent.

Simulation 2: We examine two buyer agents with "EagerSaver" and "EagerSpender" personalities.

Hypothesis 2: Since the EagerSpender agent has a higher propensity to spend, and is more eager than the EagerSaver agent, we expect the EagerSpender agent to out-compete the EagerSaver agent in instances where both agents attempt to bid in the same service. Moreover, since the EagerSpender agent does not care about the price of an item, we anticipate that it would gain higher utility in direct sales protocols.

Simulation 3: We examine three buyer agents with all three personalities; "EagerSaver", "EagerSpender", "Saver".

Hypothesis 3: We expect the Saver agent to only bid when the service price is low and therefore to remain dormant unless a service exists in the market with an item price low enough to entice the agent. We anticipate that the Saver agent would be out-competed in cases where it shares a protocol with other buyer agents (as competitors), and therefore have the lowest utility amongst other buyers.

Simulations 4–7: We extract traits from four historical Ebay auctions whose dynamics are illustrated in Figure 7. We use these traits to construct four auction-only settings composed of three to five buyer agents.

Hypotheses 4–7: We expect the bidding dynamics of the simulated agents to closely correspond to the dynamics of human bidders in the historical auctions, where the bids being placed reflect the historical trend.

5.3 Results

Below, we summarize our main observations about the first three simulations, and compare them with our initial hypotheses.

No competition: In Simulation 1, where there are no competing buyers, we observe via Figure 4 that the EagerSaver agent maintains a mean utility of one, as in each run it has three possible services to choose from with no competition. As compatible with our hypothesis, it chooses direct sales and negotiation protocols more often than auctions.

Spend vs save: In Simulation 2, we have two competing buyers (EagerSaver and EagerSpender), which are both more likely to choose direct sales and negotiation protocols than Auctions. However,

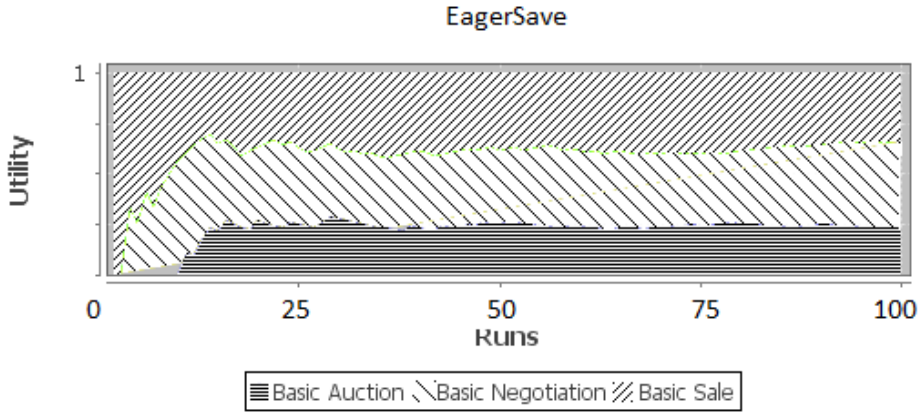


Figure 4: The stacked mean utility for EagerSaver in Simulation 1.

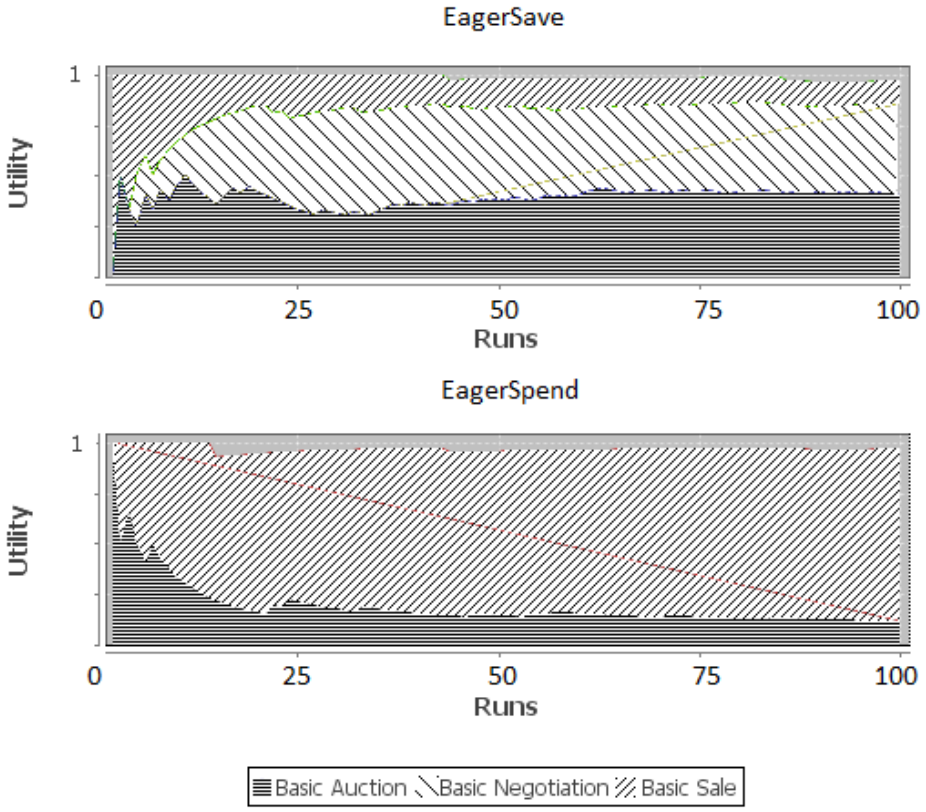


Figure 5: The stacked mean utilities for EagerSaver and EagerSpender in Simulation 2.

since EagerSpender can spend more and is highly eager in nature, we observe via Figure 5 that it out-competes EagerSaver in almost all direct sale protocols, forcing the EagerSaver agent to fulfill

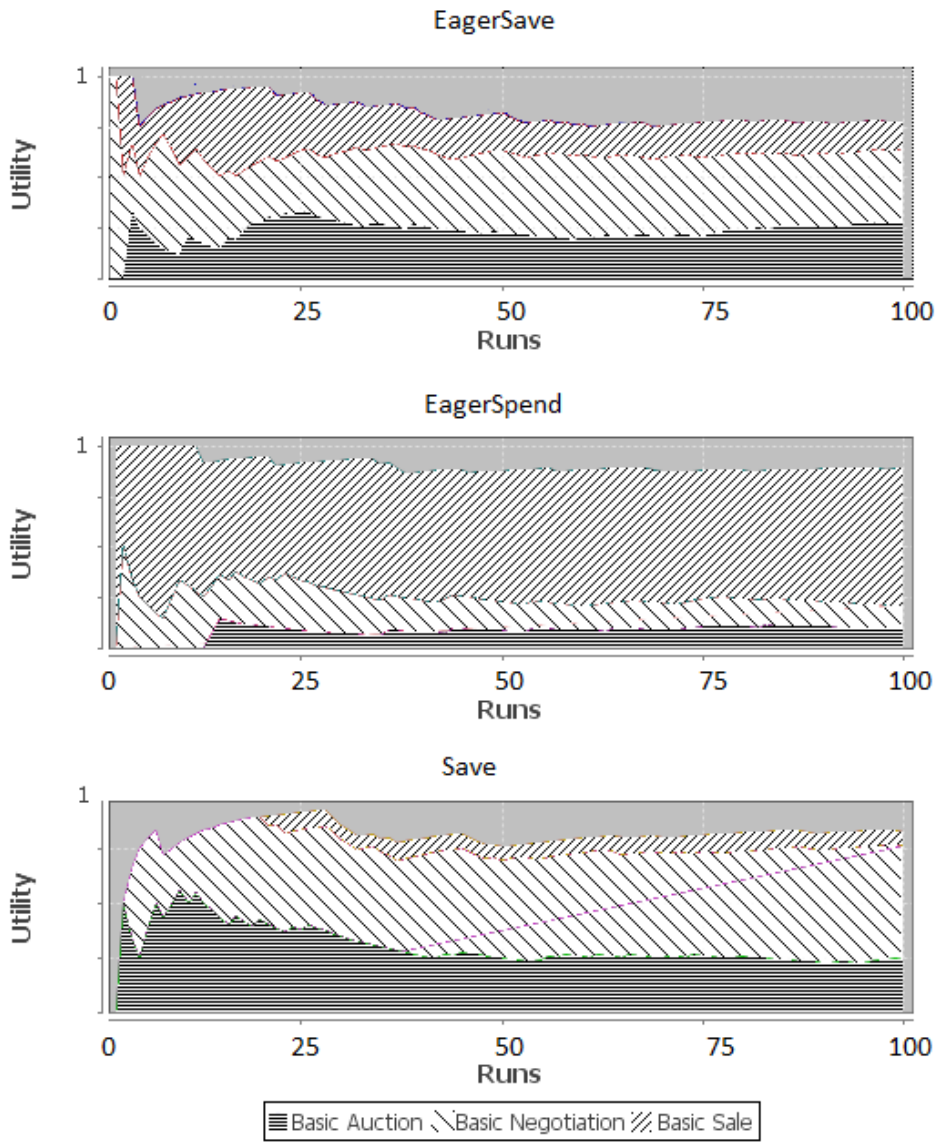
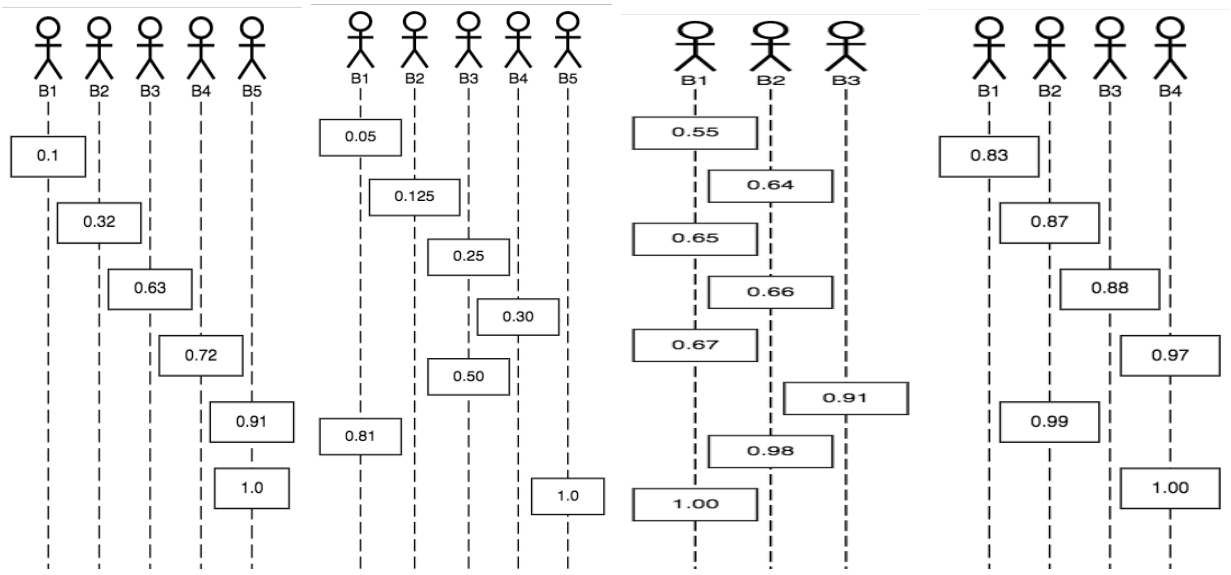


Figure 6: The stacked mean utilities for EagerSaver, EagerSpender and Saver in Simulation 3.

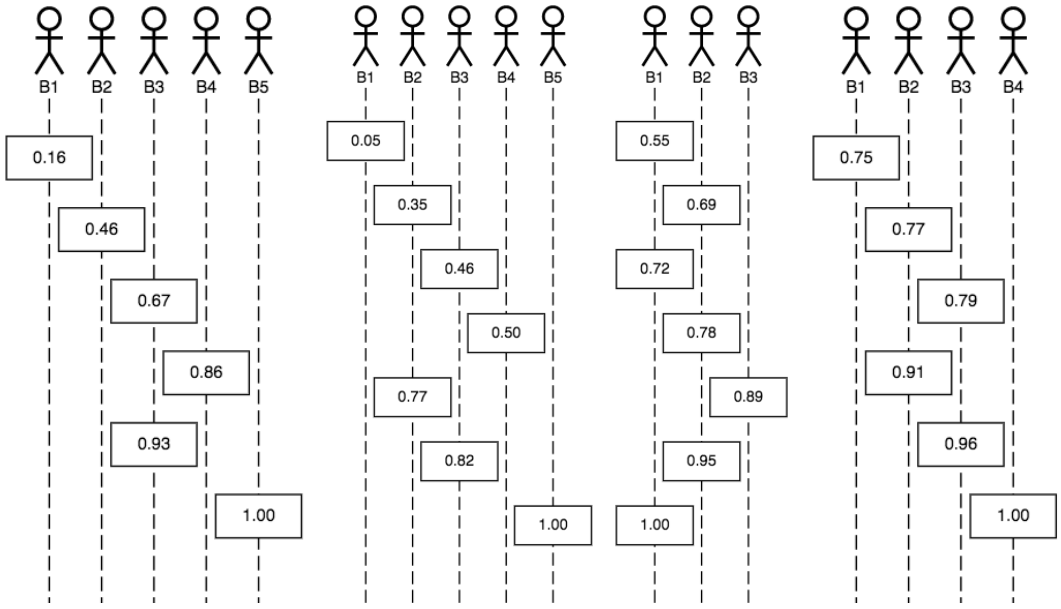
its goal mainly through negotiation and auction protocols (compatible with our hypothesis). This competition results in both agents ending the simulation with high, albeit suboptimal utility.

Effect of competition: In Simulation 3, we have all three buyer personalities in the same environment. The Saver agent is far less likely to spend money in pursuit of its goals, and is therefore more timorous than the other two agents. We observe via Figure 6 that there is significant competition between the agents, driving down the overall utilities of all agents. Surprisingly and in contrary to our initial hypothesis, the Saver agent maintains a higher mean utility than the EagerSaver agent. This is possibly due to the agent being less likely to compete in direct sales protocols against the EagerSpender agent, and instead going for negotiation protocols. This is a different strategy than



(a) Auction 1: 1641629836 (b) Auction 2: 1643136423 (c) Auction 3: 1644309984 (d) Auction 4: 1649848613

Figure 7: Four historical Ebay Auctions (with auction IDs), each lasting seven days, with normalised bids.



(a) Auction 1 (b) Auction 2 (c) Auction 3 (d) Auction 4

Figure 8: Four simulated Ebay Auctions, using agent traits mined from an analysis of each historical auction.

what the EagerSaver agent adopts. The EagerSaver agent (unsuccessfully) attempts to compete with EagerSpender on direct sale protocols.

The above results support our hypotheses that the agents with high "Spend" and "Eager" modifiers have the highest utility, and generally out-compete agents with "Save" modifiers in markets with a fixed number of protocols and agents. Moreover, we observe that competition (the inclusion of other buyers in the market) impacts the utility of buyers in closed market environments, resulting in a change of utility from 1.0 to 0.76 from Simulation 1 to Simulation 3.

Historical data: In Simulations 4–7, we find that capturing the 'Eagerness', 'Spending', and 'Saving' attributes of agents and using the 'Max Cost' constraints from bid history are sufficient to replicate the dynamics of four historical Ebay auctions (Jank & Shmueli, 2017). A comparison of Figures 7 (real auction data) and 8 (agent-based simulations) shows that our simulated buyer agents make bids that closely mimic those placed by their human counterparts. Moreover, due to the autonomy associated with agent behaviours, some agent bids show deviations from mined data, which would enable e-commerce researchers to simulate settings with varying market dynamics. Further experimentation in this direction would improve the prediction of bidding behavior in online auctions using human-based historical data, and enable the development of more sophisticated agents.

5.4 Performance Evaluation

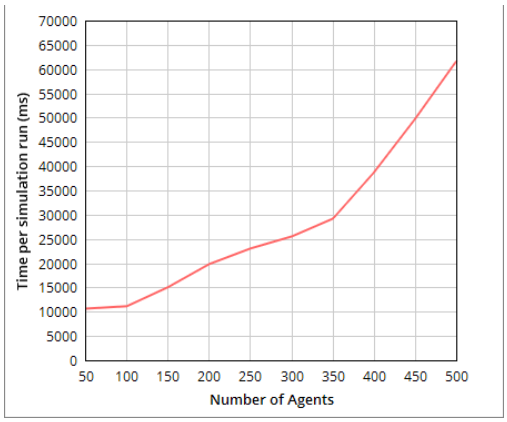
We evaluate the performance of PIRASA through a series of experiments with numbers of agents ranging from 50 to 500, incrementing by 50. We run these experiments on an i7 4770k computer with 16gb of memory running Windows 7 64-bit OS. For each experiment we run 3 sets of 10 simulation runs with an equal number of buyers and sellers, finding the average time taken for 10 runs to elapse. We then derive from this the average length of a single run. One can see from Figure 9(a) that there is a linear increase in the simulation time, up until 350 agents. This number of agents is quite significant compared to other e-commerce platforms with considerable computational requirements for the running agents.

We run a similar performance evaluation for RECON (Alrayes, Kafalı, & Stathis, 2016). Figure 9(b) shows that the average cycle time for buyer agents in RECON grow linearly over 100 runs with increasing numbers of agents. Although the simulation settings in PIRASA and RECON vary, we can see by comparison of two plots that PIRASA can support a fairly large number of concurrent agents to simulate e-commerce protocols. Moreover, note that PIRASA can support multiple protocols whereas RECON is specifically built for negotiation protocols.

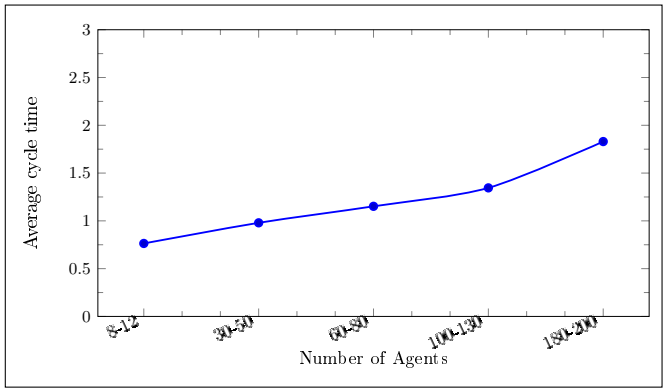
6 Related Work

In this section, we review relevant literature on agent-based simulation environments and e-commerce platforms, and compare their contributions to PIRASA.

There has been extensive research in recent years into agent negotiation in the context of marketplace simulation, usually with a focus on negotiation strategies. The e-Game (electronic Generic auction marketplace) platform (Fasli & Michalakopoulos, 2008) is a Java based, FIPA compliant platform that shares many similarities with PIRASA. Built for simulations into market infrastructure, negotiation protocols, and strategic behavior, it supports complex and dynamic auctions, facilitated by a *scheduler* agent, which is roughly analogous to the *broker* agent in PIRASA. How-



(a) The time taken per simulation run in PIRASA.



(b) Average cycle time in RECON.

Figure 9: Performance with increasing numbers of agents.

ever, e-Game does not just provide facilities for scheduling, running and conducting experiments, but enables modular implementation of auction-based market simulations. The development of new simulations in e-Game is tedious, requiring several thousand lines of code for a simple market scenario.

Another attempt at automatic agent negotiation in the context of auctions is the work of Benameur et al. (Benameur, Chaib-Draa, & Kropf, 2002), which considers a market setting with: (i) one vendor (seller) and one buyer directly negotiating; (ii) multiple vendors and one buyer are engaged in a *reverse auction*; (iii) multiple buyers and one vendor are engaged in a *classical auction*; (iv) multiple buyers and vendors trade in a market. Their approach differs from ours as they do not include *broker* agents, instead letting the vendors manage the auctions themselves. Since they do not rely on a third party agent, their auction model is generic enough to simulate each instance of their market framework. However, their implementation assumes that agents interact in accordance with the English auction protocol (open-outcry ascending) (Wooldridge, 2009), which is a major limitation when simulating complex market scenarios. As a result, one-to-one bargaining could not be simulated.

Multiagent negotiation platforms are proposed to implement and analyze automated agent strategies for negotiation. The eAgora platform (Chen, Vahidov, & Kersten, 2005) is an e-marketplace constructed for the simulation of multi-issue negotiations (a setting that cannot be easily modeled with auctions), where either the buyer or the seller can be the host. This is a difference from PIRASA. In addition, eAgora represents agent negotiation strategies as either competitive, collaborative, compromising, or accommodating, which loosely corresponds with the attributes of 'saver' and 'spender' (with different proportional values) in PIRASA. However, only the above four strategies are supported, which is a limitation when compared to the adaptive nature of the infinitely many strategies that can be constructed in PIRASA.

GENIUS (Lin et al., 2014) is a negotiation environment that implements an open architecture for heterogeneous negotiating agents. It provides a testbed for negotiating agents that includes a set of negotiation problems for benchmarking agents, a library of negotiation strategies, and analytical tools to evaluate an agent's performance. GENIUS is mainly used for evaluating bilateral negotiations, especially for agents participating in the automated negotiating agents competition

(ANAC) (Fujita et al., 2013). Williams et al. extended GENIUS to provide support for concurrent negotiations (Williams, Robu, Gerding, & Jennings, 2012). However, this extension addresses a specific experimental setup, and is not publicly accessible. Moreover, we are not aware of any work that evaluates the robustness and scalability of GENIUS when using a large number of agents.

Motivated by the current limitations in GENIUS, RECON (Alrayes et al., 2016) was developed as a robust multiagent environment for simulating concurrent negotiations. RECON is build on top of the GOLEM agent platform (Bromuri & Stathis, 2008), and supports the development of software agents (both buyers and sellers) negotiating concurrently with other agents over multiple issues. In contrast to most agent development platforms such as GENIUS, which only support imperative agents built in Java, RECON supports agents developed with declarative program. Declarative agents enable developers to specify strategies that can be transparent to a human user, in that explanations can be provided for describing why the agent has taken certain actions during a negotiation.

There are many commercial negotiation simulation environments in the market (Sim, 2002): Tete-a-Tete, Kasbah, AuctionBot, and the Fisher market. Fisher market is based on the Dutch auction protocol. The limitation with the Dutch auction is when there is one seller then the auctioneer will sell the product for the seller if the buyer reservation price has not yet reached. Whereas, PIRASA supports many protocol types depending on how many sellers and buyers are in the market, and whether the goals of the seller and buyer are to maximize profit or to prefer a quick transaction. AuctionBot supports many auction types including the Dutch auction. The same limitation for Fisher market applies to AuctionBot, it is not suitable when the sellers want to negotiate quickly or if there is one agent in the market. On the other hand, Tete-a-Tete and Kasbah offers one kind of negotiation protocol, whereas PIRASA offers three kind of protocols, which can further be customised and extended. One novelty that PIRASA provides is that we are not limited to one type of protocol. A variety of protocols, including Dutch auctions, can be integrated into PIRASA using the customizable protocol attributes, as well as adding new attributes.

7 Conclusions and Future Work

In this paper, we have presented PIRASA: an agent-based platform for simulating e-commerce protocols. It allows agents to determine which protocol is more beneficial via experimentation in different settings. It supports customization of agent attributes, which govern the agent's behavior. This enables to simulate realistic e-markets where buyers compete with other to buy items from seller. Our attempt is a first to do this in real-time.

PIRASA supports basic goals and constraints for the agents, e.g., a deadline to purchase a specific item. Future work could result in hierarchical trees of predicates to allow agents to retain highly complex goals. Goals could also be extended to be domain specific, conforming to a predefined ontology. In the current framework, we have focused on buyers and have not implemented complicated seller strategies. Having goals for the sellers as well would lead to more realistic e-markets.

Having a depreciation or appreciation modifier for items would add a new layer into PIRASA, as it would allow buyers to be more strategic when choosing services to bid on, i.e., a Saver agent could choose to fulfill a goal with a lower quality yet cheaper item, as opposed to spending more for a better item. This would also tie into seller strategy, as the quality of an item could partially dictate the protocol they use with which to sell it. For example, when trying to sell a slightly depreciated item, such as a car which has had a previous owner, there is a greater scope for negotiation. Similarly,

low quality items, such as an old sofa, could either be negotiated or auctioned.

There might be other protocol types that can be supported by modifying the protocol parameters. A Dutch auction (also known as a clock auction or an open-outcry descending-price auction) is the idea that a seller tries to sell an item at a very high price, and slowly over time lowers that price until some buyer is willing to pay and obtain the item. This is referred to as an auction because it tends to provide the market ceiling price for a seller.

We plan to extend our work in the following directions:

- We have evaluated protocols from the buyer's point of view. In a more realistic setting, a trader agent might act both as a buyer and a seller in multiple competitive markets. Investigation of such markets, as well as taking into account the violations of agents' norms, can help understand how metrics such as social welfare evolve in those markets.

- Performing statistical tests and understanding the connection between agent traits and protocol properties more closely would be helpful to determine whether a protocol is significantly more beneficial to an agent over the others.

- PIRASA can be extended to include additional multiagent protocols such as argumentation (Gao, Toni, Wang, & Xu, 2016; Kökciyan, Yaglikci, & Yolum, 2017) for resolving conflicts among trading agents as well as between agents and their users. Such agents should learn from previous encounters with other agents as well as take into account user's feedback on previous choices. Whereas such interactions are helpful among the agent and its user, they introduce privacy concerns, e.g., what amount of information should users share with their agents to maximize their utilities, and how would this information propagate in the market? In its current form, an agent in PIRASA does not reveal its users' preferences to other agents. The agent only simulates the potential interactions as they would be observed in a real transaction. Agent-based privacy solutions (Baarslag et al., 2017; Kafalı, Ajmeri, & Singh, 2016) can be employed to address information disclosure concerns when the agent interacts with other agents as well as its user.

- PIRASA can be extended to enable the participation of human agents as part of the simulations to capture more realistic e-commerce settings. Such humans agents would not be autonomous, and would simply follow orders from a human operator. Crowdsourcing studies can be conducted to evaluate the interactions among intelligent agents and human users.

References

Akula, V., & Menascé, D. A. (2004). An analysis of bidding activity in online auctions. In *Proceedings of the international conference on electronic commerce and web technologies* (pp. 206–217).

Alrayes, B., Kafalı, Ö., & Stathis, K. (2016). Recon: A robust multi-agent environment for simulating concurrent negotiations. In *Recent advances in agent-based complex automated negotiation* (pp. 157–174). Springer International Publishing.

Alrayes, B., Kafalı, Ö., & Stathis, K. (2017, Oct). Concurrent bilateral negotiation for open e-markets: the conan strategy. *Knowledge and Information Systems*. Retrieved from `https://doi.org/10.1007/s10115-017-1125-2` doi: 10.1007/s10115-017-1125-2

Alt, R., & Zimmermann, H.-D. (2016). Electronic markets on electronic markets in education. *Electronic Markets*, *26*(4), 311–314. Retrieved from http://dx.doi.org/10.1007/s12525-016-0237-y doi: 10.1007/s12525-016-0237-y

Baarslag, T., Alan, A. T., Gomer, R., Alam, M., Perera, C., Gerding, E. H., & schraefel, m. (2017). An automated negotiation agent for permission management. In *Proceedings of the 16th conference on autonomous agents and multiagent systems (aamas)* (pp. 380–390).

Bataineh, A. S., Bentahar, J., Menshawy, M. E., & Dssouli, R. (2017). Specifying and verifying contract-driven service compositions using commitments and model checking. *Expert Systems with Applications*, *74*, 151–184.

Baumer, C., Breugst, M., Choy, S., & Magedanz, T. (1999). Grasshopper: A universal agent platform based on OMG MASIF and FIPA standards. In *First international workshop on mobile agents for telecommunication applications* (pp. 1–18).

Bellifemine, F., Poggi, A., Rimassa, G., & Turci, P. (2000). An object-oriented framework to realize agent systems. In *Woa workshop: From objects to agents* (p. 52-57).

Benameur, H., Chaib-Draa, B., & Kropf, P. (2002). Multi-item auctions for automatic negotiation. *Information and Software Technology*, *44*(5), 291–301.

Boella, G., & van der Torre, L. (2008, March). Institutions with a hierarchy of authorities in distributed dynamic environments. *Artificial Intelligence and Law*, *16*(1), 53-71.

Bromuri, S., & Stathis, K. (2008). Situating cognitive agents in GOLEM. In D. Weyns, S. Brueckner, & Y. Demazeau (Eds.), *Engineering environment-mediated multi-agent systems* (Vol. 5049, pp. 115–134). Springer Berlin Heidelberg.

Chen, E., Vahidov, R., & Kersten, G. E. (2005). Agent-supported negotiations in the e-marketplace. *International Journal of Electronic Business*, *3*(1), 28–49.

Chesani, F., Mello, P., Montali, M., & Torroni, P. (2013, July). Representing and monitoring social commitments using the event calculus. *Autonomous Agents and Multi-Agent Systems*, *27*(1), 85–130.

Dechesne, F., di Tosto, G., Dignum, V., & Dignum, F. (2013). No smoking here: Values, norms and culture in multi-agent systems. *Artificial Intelligence and Law*, *21*(1), 79–107.

Fasli, M., & Michalakopoulos, M. (2008). e-game: A platform for developing auction-based market simulations. *Decision Support Systems*, *44*(2), 469–481.

Fujita, K., Ito, T., Baarslag, T., Hindriks, K., Jonker, C., Kraus, S., & Lin, R. (2013). The second automated negotiating agents competition (anac2011). In T. Ito, M. Zhang, V. Robu, & T. Matsuo (Eds.), *Complex automated negotiations: Theories, models, and software competitions* (Vol. 435, p. 183-197). Springer Berlin Heidelberg.

Gao, Y., Toni, F., Wang, H., & Xu, F. (2016). Argumentation-based multi-agent decision making with privacy preserved. In *Proceedings of the 15th conference on autonomous agents and multiagent systems (aamas)* (pp. 1153–1161).

Governatori, G. (2013). Business process compliance: An abstract normative framework. *Information Technology*, *55*(6), 231–238.

Harris, M., & Raviv, A. (1981). Allocation mechanisms and the design of auctions. *Econometrica*, *49*(6), 1477-1499.

Jank, W., & Shmueli, G. (2010). *Modeling online auctions* (Vol. 91). John Wiley & Sons.

Jank, W., & Shmueli, G. (2017). *Modeling online auctions datasets.* (Snapshot taken on October 2017. http://www.modelingonlineauctions.com/datasets)

Kafalı, Ö. (2012). *Automated reasoning on exceptions in commitment-based multiagent systems*

(Unpublished doctoral dissertation). Boğaziçi University.

Kafalı, Ö., Ajmeri, N., & Singh, M. P. (2016, September). Revani: Revising and verifying normative specifications for privacy. *IEEE Intelligent Systems*, *31*(5), 8–15.

Kafalı, Ö., Singh, M. P., & Williams, L. (2016, September). Nane: Identifying misuse cases using temporal norm enactments. In *Proceedings of the 20th ieee international requirements engineering conference (re)* (pp. 136–145). Beijing: IEEE Computer Society.

Kafalı, Ö., & Torroni, P. (2012). Exception diagnosis in multiagent contract executions. *Annals of Mathematics and Artificial Intelligence*, *64*(1), 73-107.

Kafalı, Ö., & Yolum, P. (2016, April). Pisagor: A proactive software agent for monitoring interactions. *Knowledge and Information Systems*, *47*(1), 215–239.

Kökciyan, N., Yaglikci, N., & Yolum, P. (2017, June). An argumentation approach for resolving privacy disputes in online social networks. *ACM Transactions on Internet Technology (TOIT)*, *17*(3), 27:1–27:22.

Kowalski, R., & Sergot, M. (1986). A logic-based calculus of events. *New Generation Computing*, *4*(1), 67–95.

Lin, R., Kraus, S., Baarslag, T., Tykhonov, D., Hindriks, K. V., & Jonker, C. M. (2014). GENIUS: An integrated environment for supporting the design of generic automated negotiators. *Computational Intelligence*, *30*(1), 48–70.

Lukasiewycz, M., Glaß, M., Reimann, F., & Teich, J. (2011). Opt4J - A Modular Framework for Meta-heuristic Optimization. In *Proceedings of the genetic and evolutionary computing conference (gecco 2011)* (pp. 1723–1730). Dublin, Ireland.

Luke, S., Cioffi-Revilla, C., Panait, L., & Sullivan, K. (2004). Mason: A new multi-agent simulation toolkit. In *Proceedings of the 2004 swarmfest workshop* (Vol. 8).

Montali, M., Calvanese, D., & De Giacomo, G. (2014). Verification of data-aware commitment-based multiagent system. In *Proceedings of the international conference on autonomous agents and multi-agent systems (aamas)* (pp. 157–164). Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Parsons, S., Rodriguez-Aguilar, J. A., & Klein, M. (2011, February). Auctions and bidding: A guide for computer scientists. *ACM Comput. Surv.*, *43*(2), 10:1–10:59. Retrieved from `http://doi.acm.org/10.1145/1883612.1883617` doi: 10.1145/1883612.1883617

Sein, M. K., Henfridsson, O., Purao, S., Rossi, M., & Lindgren, R. (2011, March). Action design research. *MIS Quarterly*, *35*(1), 37–56.

Sim, K. M. (2002). A market-driven model for designing negotiation agents. *Computational Intelligence*, *18*(4), 618–637.

Singh, M. P. (2013, December). Norms as a basis for governing sociotechnical systems. , *5*(1), 21:1–21:23.

Spiekermann, S., Böhme, R., Acquisti, A., & Hui, K.-L. (2015). Personal data markets. *Electronic Markets*, *25*(2), 91–93. Retrieved from `http://dx.doi.org/10.1007/s12525-015-0190-1` doi: 10.1007/s12525-015-0190-1

Von Wright, G. H. (1999, March). Deontic logic: A personal view. *Ratio Juris*, *12*(1), 26–38.

Williams, C. R., Robu, V., Gerding, E. H., & Jennings, N. R. (2012, August). Negotiating concurrently with unknown opponents in complex, real-time domains. In *20th european conference on artificial intelligence* (Vol. 242, pp. 834–839).

Wooldridge, M. (2009). *An introduction to multiagent systems* (2nd Edition edition, Ed.). John Wiley & Sons.

Xu, H., & Shatz, S. M. (2003). Adk: An agent development kit based on a formal design model for multi-agent systems. *Automated Software Engineering*, *10*(4), 337–365.