

Laboratory assignment

Component 3

Authors: Liviu-Ștefan Neacșu-Miclea, Răzvan-Gabriel Petec

Group: 246/2

April 16, 2025

1 Conceptual model using PAGES

The PAGES framework stands for Percepts, Actions, Goals, Environment, and State. It is useful for modeling agent-based systems at the conceptual level. Table 2 showcases a PAGES description for each of the agents acting in the environment.

Environment consists of the shared digital stock market, which includes a set of tradable companies or stocks, a price-setting mechanism influenced by agents' trades and possibly external events (e.g., company announcements), and facilities for tracking current prices of stocks, recording historical prices and trade activity storage.

State of the environment contains the following properties:

- **stockPrices:** Dict<String, Number> = the current price of each stock
- **priceHistory:** Dict<String, Number> = a list of previous prices for each stock
- **tradeHistory:** Dict<String, String[]> = a log of trades made (by which agents, what they did)

Table 1 presents additional state retained by each agent besides the stock market data.

Agent	State
TrendTrader (TT)	
InfoTrander (IT)	Trust levels for each company
Company (C)	Last project info
Regulator (R)	

Table 1: Additional agent states particularities

2 Properties of the environment

Table 3 shows each agent's accessibility in the environment.

The environment properties are detailed in table 4.

3 Application design

Figure 1 showcases the class diagram of the MAS.

Trader is the abstract class, extended by TrendTrader and InfoTrader.

Company emits project events via a PubSubBus (publisher-subscriber).

Market processes all trade actions and updates the state.

Agent	Percepts	Actions	Goals
TrendTrader (TT)	Historical stock prices (last k prices per company).	Buy, Sell, Idle.	Maximize profit by exploiting price trends (buy low, sell high).
InfoTrader (IT)	ProjectEvent messages from companies.	Buy, Sell, Idle.	Maximize profit by trusting companies with successful projects.
Company (C)	None (internally driven).	Emit ProjectEvent (risk, succeeded, failed).	Introduce market signals and influence volatility.
Regulator (R)	Trade history.	Adjust stock prices (up/down, bounded by a minimum price).	Maintain market stability.

Table 2: Agents description using PAGES (Percepts, Actions, Goals)

Agent	Accessibility	Details
TrendTrader (TT)	inaccessible	Stock market partially (stock prices, transaction history).
InfoTrander (IT)	inaccessible	Project event stream, stock market partially.
Company (C)	inaccessible	Not reactive to external stimuli.
Regulator (R)	accessible	Global stock transaction records

Table 3: Accessibility

Property	Type	Explanation
Determinism	Non-deterministic	Stock prices and company events can change unpredictably.
Dynamics	Dynamic	The environment changes independently via companies and trades.
Discrete vs. Continuous	Discrete	Time steps, events, and trades happen in discrete intervals.
Shared Environment	Yes	All agents affect and react to a common stock market.
Markovian	Yes	The next state only depends on the current state of the environment.

Table 4: Description of the environment

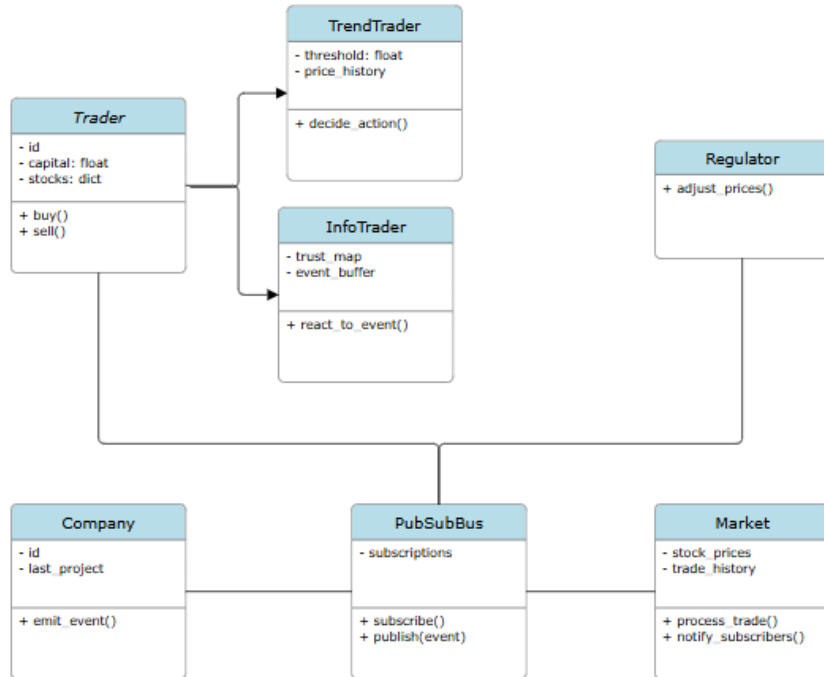


Figure 1: Class diagram

Regulator monitors the market state and updates prices.

PubSubBus manages asynchronous delivery of project events to InfoTraders.

4 Communication and interaction model

The system uses a Publish-Subscribe (Pub/Sub) pattern for asynchronous, event-driven communication between agents.

Component	Communicates With	Via	Message Type
Company	InfoTrader(s)	PubSubBus	ProjectEvent(C-j, status)
Trader	Market	Direct method call	buy(), sell()
Regulator	Market	Observes	Reads trade_history
Market	All agents	Centralized store	Stock prices

Table 5: Communication details

In the interaction flow, at each time step, a Company randomly emits project events using PubSubBus. Then, InfoTrader updates trust and possibly trades based on events and TrendTrader analyzes price trends and possibly trades. All trades go through the Market (environment), updating trade history and agent portfolios. The Regulator observes trade history and adjusts stock prices. Finally, Stock prices are updated for the next round.