University of Tehran
School of Electrical and Computer Engineering

Software Engineering 1

Spring 2024

Introduction to the Matching Engine Domain

Introduction

Matching Engine is a core component of the stock exchange market which is responsible for receiving orders from market players including brokers and shareholders, matching them against each other, and creating trades. This document aims to explain the Matching Engine, its components, and how it works. The scope covered in this document is limited to this term's project (TinyME). The document is based on the Chronicle Matching Engine Technical Report. Please note that in some parts of the domain, our specification may vary from the Chronicle's.

Main Entities

This section describes the main domain entities in a Matching Engine.

Security

Security is stock that is traded at a stock exchange market which is identified by an ID called "**ISIN"**. For example, there is stock with ISIN US0378331005 which is the Apple stock. Each security has a specific "**tick size"** which determines the price change steps for the orders for this security. Also, there is a "**lot size"** for each security that determines the quantity steps for the orders. Every time an order is placed to buy or sell a security, the order's quantity must be a multiple of the security's lot size and the order's price must be a multiple of the security's tick size.

An important aspect of security is its order book. The "**order book"** consists of separate lists for buy and sell orders and keeps the open orders in it. Orders are sorted in each list based on their price (high to low for buy orders and low to high for sell orders). Orders with the same price are sorted based on the time they entered the order book.

Trade

Trade is the outcome of the matching of two orders against each order. Each trade consists of a buy and sell order and the price and quantity of the trade. The quantity of the trade is the minimum quantity of its buy and sell orders. The price of the trade is a point between the price of buy and sell orders. This point is determined based on the matching algorithm and the properties of the orders participating.

Order

Order is the request sent from market players to indicate that they want to buy/sell some amount (quantity) of a stock. The main characteristics that define the behavior of an order are its type, time enforcements and minimum execution quantity. The list of order types (within the scope of TinyME) is as follows:

Type	Description
Limit	Executes when there are orders with prices at or better than the price that is specified by this order. Any remainder of the order is added to the order book.
Iceberg	An order which has both a quantity and a “disclosed quantity”. Upon entry it behaves as a limit order but when it's added to the order book, only the disclosed quantity is visible and may be matched with other orders. The disclosed quantity is initialized to “peak size”, and is decreased as the order is matched with the incoming orders of the opposite side. Once the displayed quantity is reached to zero, the order is removed from the order book, replenished (its displayed quantity set to the minimum of the remaining quantity and the peak size), and added again to the order book.
Stop Limit	A limit order that remains inactive until the last traded price reaches a “stop price” (specified in the order). At this point it enters the matching cycle as a normal limit order with a price that can be different from the stop price.

An order has a Time-In-Force property, which determines the times at which the order is valid. The options for this property are as follows.

Type	Description
Day	Order expires at the end of the current trading day.
Good Till Cancel (GTC)	Order remains in the system till the time that user manually cancels the order.
Good Till Date (GTD)	Order expires at the end of the specified trading day.
Good Till Time (GTT)	Order expires at the specified time.
Immediate or Cancel (IOC)	Order is executed upon entry and any remainder expires immediately.

Fill Or Kill (FOK)	Order is executed upon entry. If there is any remainder after the matching, all the trades are rolled back and order expires completely.

Also, **“minimum execution quantity”** (MEQ) defines the minimum amount of quantity of the order that must be matched upon entry to the system (before entering the order book). If this condition is not met, no trades are created and the order is rejected.

Broker

A “**broker**” is a stakeholder that can communicate with the matching engine. In many markets, individuals do not directly communicate with the matching engine to send orders and receive the responses from the matching engine. In such markets, brokers are responsible for providing the required infrastructure for their customers to place orders. As the representatives of their customers, the brokers are needed to have enough “credit” to buy shares, and earn credit after selling shares. The details of the credit management are explained in the class.

Handling Broker’s Credit

When a new buy order enters the system, it may match with some queued sell orders and create a number of trades. For each such trade, the credit of the buying broker is decreased by the value of the trade (TradeQuantity * TradePrice). If the buy order has a remainder after initial matching, it enters the order book. In this case, the broker’s credit is decreased by RemainingQuantity * OrderPrice. **If at any time during this process, the buyer’s credit becomes negative, the incoming order is rejected and all of the trades are rolled back.** On the other hand, every time a sell order is matched, the selling broker’s credit is increased by TradeQuantity * TradePrice.

With the same logic, the buyer’s credit must be correctly updated in case a queued buy order is amended or canceled.

Shareholder

Shareholders are the individuals that own security shares. They place buy or sell orders through the brokers. To sell a certain quantity of shares, the shareholder must own that amount at least. This is checked upon receiving a “new” or “amend” request on a sell order. If this check fails, the request is rejected.

Processing Pipeline

The processing pipeline indicates the cycle that each request goes through in the matching engine. It consists of the following steps:

- Validation
- Matching
- Publishing the results

This section describes the types of the requests that can be received, explains each step, and specifies the outputs of the matching.

Request Types

The types of the request that matching engine accepts are:

- New: to enter a new order to the system
- Cancel: to remove an existing order in the order book
- Amend: to change the state of an existing order in the order book

Upon amending an order only the following properties can be changed:

- Price
- Stop Price (for Stop Limit orders)
- Quantity
- Peak Size (for Iceberg orders)
- Time-In-Force

Also, amending an order can lead to it losing its priority in the order book. The following cases leads to losing priority when an order is amended:

- Increasing Quantity or Peak Size
- Changing the Price

When an order loses its priority, the updated order is removed from the order book and is treated as a new order. Note that this may cause trades to happen.

Validation

There are two types of validation. Structural validations include

- The price and quantity of the orders must be positive.
- The security, broker, and shareholder of the orders must be known.
- The peak size of the iceberg orders must be positive and less than the total quantity.
- Amend requests may only change certain attributes (as described above).

Business validations include tick and lot size checks, credit checks, and shareholder ownership checks.

Matching

After an order is validated, it enters the matching loop. The matching algorithm defines what happens to the incoming order according to the order book. There are two matching algorithms.

Continuous Matching

In continuous matching, when an order enters the matching loop, it is matched against the top order of the opposite side. This loop continues until the incoming order is completely matched or the queue of the opposite side is empty. Rule of matching for buy and sell orders is as follows.

- A buy order matches with any sell order with a price less than or equal to the price of the buy order from lowest sell price to the highest. The price of the trade is set to the price of the sell order.
- A sell order matches with any buy order with a price greater than or equal to the price of the sell order from highest buy price to the lowest. The price of the trade is set to the price of the buy order.

After the matching of an order is finished, the last traded price is checked and if any stop limit order must be activated, it is treated as a new order and enters the matching process.

Auction Matching

In auction matching, for new and update requests no trade happens and the orders are put directly in the order book. Cancel requests are processed the same as before and lead to removing an order from the order book if present.

In auction matching, there is an additional action which is the **“opening action”**. This action may be scheduled for a specified time or happen by the command of the market manager. Upon this action, an **“indicative opening price”** (IOP) is calculated based on all of the orders in the order book (the detail of the calculation is described next). Then, all of the orders that can be matched with this price are selected. Finally, the selected orders are matched against each other and a list of trades is created. Note that the price of the trades are all the same as IOP, so it does not matter in what order the orders are matched.

Calculating IOP

To be completed.

Publishing the results

After the process matching is completed, all the results of the matching are published. The result of matching is a set of events that are published publicly or only for the related parties. The public events are Trade events which contain the price and quantity of the trade. There are two private events. One is the execution result of the order which can be accepted or rejected. The second private event is the execution report event which contains the list of trades created as the result of execution of the party order.