

Hệ thống giao dịch tự động CFP

Hệ thống cần một giao diện có khả năng nhập các thông tin để chuẩn bị vào lệnh. Các thông tin bao gồm tên lệnh, mô tả lệnh (tùy chọn), hành động mua hoặc bán, giá trên và giá dưới để xét khung giá của khối lệnh, giá chốt lời và giá cắt lỗ, kích thước lô, ký hiệu của loại tài sản giao dịch (Symbol), ngày hết hạn để hệ thống có thể tự hủy khối lệnh.

Khi thêm lệnh, các khối lệnh khi vừa thêm từ UI sẽ được thêm vào hàng chờ trên hệ thống và có thuộc tính trạng thái biểu thị tính chất chờ của khối lệnh. Khối lệnh có trạng thái chờ sẽ nằm trên hàng chờ và chờ đợi tới khi các điều kiện tương ứng của khối lệnh đó được thỏa mãn. Các khối lệnh được thỏa mãn khi các thông tin cung cấp đáp ứng được các tiêu chí sau:

1. Với Charting System (hệ thống biểu đồ nói chung, không gộp khuôn trong 1 hệ thống biểu đồ nào), các chỉ báo (indicator) như MACD, RSI và các khung thỏa mãn các điều kiện sau:
 - a. Nếu khối lệnh có hành động là BUY:
 - MACD của M12 cắt lên lần đầu và xanh đậm lần đầu sau xanh lợt.
 - MACD của M75 không đỏ đậm
 - MACD của H4 không đỏ đậm
 - b. Nếu khối lệnh có hành động là SELL:
 - MACD của M12 cắt xuống lần đầu và đỏ đậm lần đầu sau đỏ lợt.
 - MACD của M75 không xanh đậm
 - MACD của H4 không xanh đậm
2. Với Pricing System (hệ thống lấy giá, không gộp khuôn trong bất kỳ hệ thống nào) phải thỏa các điều kiện sau:
 - Giá tại sản tại thời điểm xét giá phải nằm trong khoảng từ giá dưới tới giá trên.

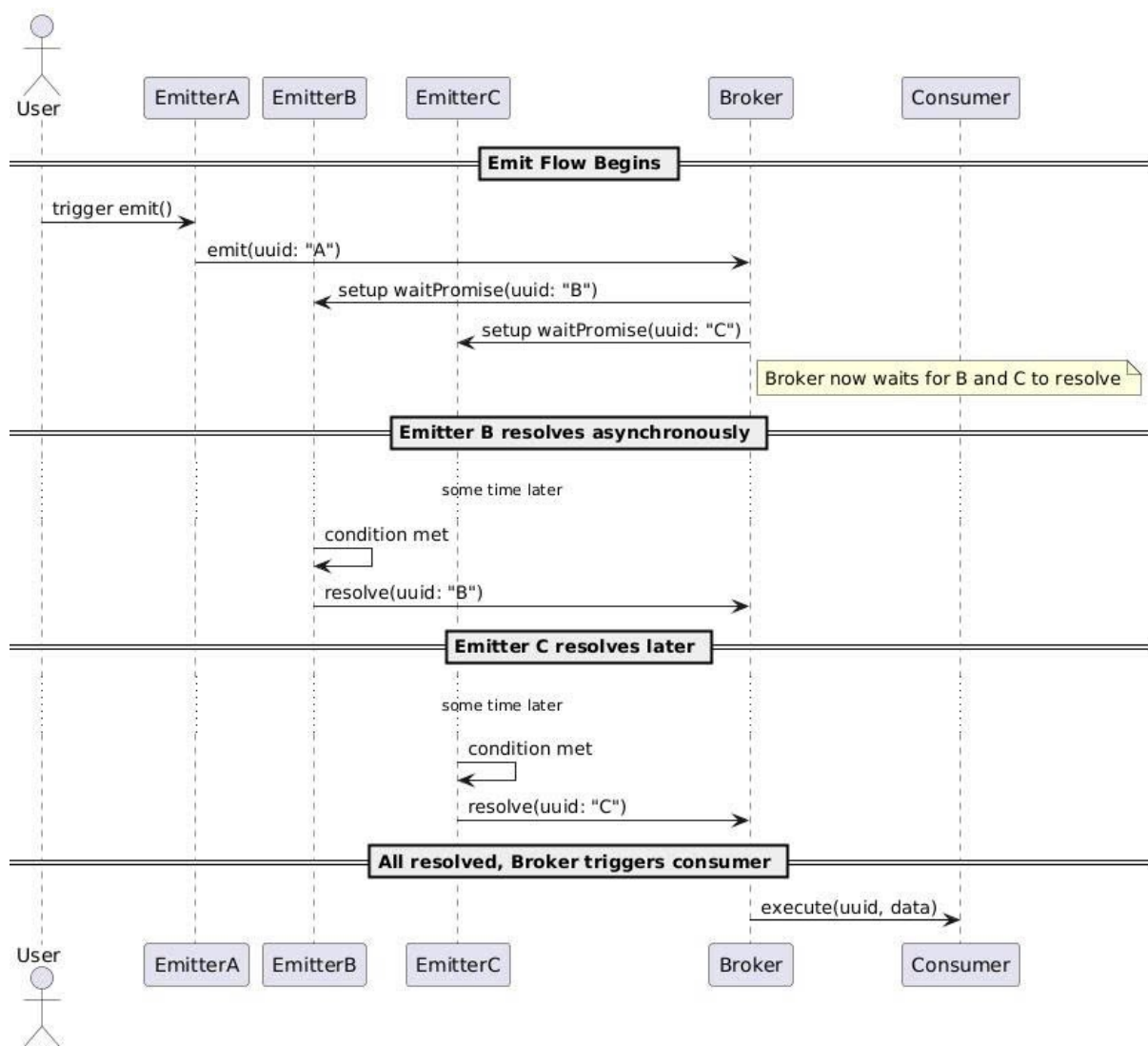
Thông thường, các hệ thống này sẽ là một và không phân tách ra, khi ta xét giá trên Meta Trader, ta có thể tự phân tích và vẽ biểu đồ cùng một lúc. Điều này đảm bảo được tính nhất quán của các giá trị tiền tệ. Tuy nhiên, trong quá trình phân tích, tôi nhận ra rằng các biểu đồ và charting của Meta Trader có sự thay đổi so với tín hiệu biểu đồ trên TradingView. Tuy anh Quang sử dụng Meta Trader là nền tảng giao dịch chính nhưng lại dùng Trading View là nền tảng xem thông tin biểu đồ. Nên khi giao dịch, anh tin vào các thông tin mà Trading View mang lại thay vì các thông tin do Meta Trader. Vì thế khi tại cùng một thời điểm giá tiền (thời gian), thuật toán MACD trên Meta Trader cho ra kết quả khác thì TradingView lại cho ra kết quả khác. Điều này xảy ra do sự chênh lệch nhỏ về giá tiền giữa hai sàn giao dịch. Để khắc phục tình trạng này, chúng ta có thể sử dụng nhiều hệ thống/nền tảng khác nhau làm nguồn thông tin tin cậy khác nhau cho từng loại tiêu chí/điều kiện. Đối với hệ thống biểu đồ, chúng ta có thể tin tưởng tín hiệu từ TradingView gửi qua Webhook. Đối với hệ thống giá tiền, ta có thể tin tưởng vào Meta Trader. Ngoài ra, để đáp ứng được với sự đa nguồn và bất đồng bộ này, tôi có tạo ra một mô hình “hướng sự kiên dựa trên mong đợi”.

Để bắt đầu, chúng ta tạo ra hai Emitters là A và B sau đó đăng ký vào Broker. Khi nguồn A (có gắn Emitter A) thông báo tin rằng điều kiện đã đạt, Broker sẽ bỏ nguồn B vào lời hứa (Promise trong typescript) và bắt B hứa rằng trong thời gian n giây, B sẽ gửi lại thông báo tiêu chí hợp lệnh. Khi nào phiên (Promise) hứa hoàn thành?

Nếu cả hai đều gửi trong khoảng thời gian n, Broker sẽ kết luận rằng phiên hứa vừa rồi thành công và tiếp tục thông báo cho các Consumer. Điều này khiến các Consumer không cần biết Emitter là ai và các nguồn (emitters) có thể giao tiếp với Broker một cách bất đồng bộ mà không sợ ai thông báo trước, ai thông báo sau, điều này có nghĩa nếu A là TradingView thông báo tín hiệu hợp lệnh trước thì B với Meta Trader thông báo sau và ngược lại hoàn toàn bình thường.

Ngược lại, nếu trong khoảng thời gian n (có thể tùy chỉnh được trước khi bắt đầu phiên) mà cả hai đều không hoàn tất, phiên sẽ bị hủy và dữ liệu về các lần thông báo (kịch hoạch) trước đều bị bỏ.

Mô hình có thể vận hành trên nhiều Emitters và Consumers, không có định số lượng Emitters và Consumers.



Trong mô hình trên, User có thể hiểu là các nền tảng sử dụng Emitter như Trading View, Meta Trader. Khi các tiêu chí chúng ta đặt cho các nền tảng đó thỏa thì các nền tảng đó sẽ kích hoạt (trigger) các Emitter đi kèm. Điều này giúp cho các nền tảng có khả năng giao tiếp với Broker.

Một đoạn code hướng dẫn sử dụng thư viện *cooter* này:

```
1  import { Broker, Emitter, Consumer } from "./src/core";
2
3  const broker = new Broker({});
4
5  const emitterA = new Emitter({
6    uuid: "Emitter A"
7  });
8  const emitterB = new Emitter({
9    uuid: "Emitter B"
10 });
11 const emitterC = new Emitter({
12   uuid: "Emitter C"
13 });
14
15 const consumerA = new Consumer({});
16 const consumerB = new Consumer({});
17 const consumerC = new Consumer({});
18
19
20 broker.registerEmitter([emitterA, emitterB, emitterC]);
21 broker.registerConsumer([consumerA, consumerB, consumerC]);
22
23 (async () => {
24   emitterA.emit();
25
26   // These could be resolved later on their own condition
27   setTimeout(() => emitterB.resolve(), 300);
28   setTimeout(() => emitterC.resolve(), 200);
29 })();
30
```

Ctrl+L to chat, Ctrl+K to generate

Trong hình trên, ta có thể thấy rằng emitterA được định sẵn sẽ kích hoạt Broker và mở phiên. Lúc này trong Broker được khai báo tại dòng số 3 sẽ bỏ các emitterB và emitterC được đăng ký tại dòng 20 vào Promise. Khi chương trình chạy, emitterB sẽ resolve (hay kích hoạt tương tự A nhưng kiểu kích hoạt khác nhau nên dùng tên hàm khác nhau) trong thời gian là 300ms và emitterC trong 200ms là thành công vì Broker hiện tại được đặt mặc định là 500ms.

Khi trong hàng chờ, lệnh sẽ biến từ trạng thái AWAITING thành TRADING biểu thị lệnh đã được vào. Lúc này Meta Trader cần cho ta một id để nhận biết lệnh này với lệnh kia, từ đó lập một danh sách các lệnh đã được vào. Các lệnh đã được vào có thể thông qua id do Meta

Trader cung cấp để thay đổi, ta dùng id đó, kèm theo các giá trị mới và đặt order lại, lúc này ta có thể chỉnh sửa TakeProfits và StopLoss dựa trên mong muốn.

Khi giám sát các lệnh, ta có thể nhận biết được lệnh nào đã ngừng và ngừng với trạng thái gì. Nếu lệnh ngừng với trạng thái `mt5.TRADE_RETCODE_DONE_SL`, lệnh đã lỗ, ngược lại với trạng thái `mt5.TRADE_RETCODE_DONE_TP`, lệnh đã lời. Chúng ta có thể phát triển một panel quản lý các lệnh dựa vào đây.

Danh sách một số các trạng thái giao dịch phổ biến:

<code>mt5.TRADE_RETCODE_DONE</code>	10009	Lệnh đã xử lý thành công
<code>mt5.TRADE_RETCODE_DONE_PARTIAL</code>	10010	Lệnh xử lý thành công một phần
<code>mt5.TRADE_RETCODE_ERROR</code>	10019	Lỗi chung
<code>mt5.TRADE_RETCODE_REQUOTE</code>	10004	Giá đã thay đổi, cần báo giá lại
<code>mt5.TRADE_RETCODE_REJECT</code>	10006	Lệnh bị từ chối
<code>mt5.TRADE_RETCODE_CANCEL</code>	10007	Lệnh bị hủy bởi server
<code>mt5.TRADE_RETCODE_PLACED</code>	10011	Lệnh chờ đã được đặt
<code>mt5.TRADE_RETCODE_DONE_SL</code>	10012	Lệnh đã đóng bởi Stop Loss
<code>mt5.TRADE_RETCODE_DONE_TP</code>	10013	Lệnh đã đóng bởi Take Profit
<code>mt5.TRADE_RETCODE_PENDING</code>	10014	Lệnh đang trong trạng thái chờ xử lý
<code>mt5.TRADE_RETCODE_NOT_ENOUGH_MONEY</code>	10015	Không đủ tiền
<code>mt5.TRADE_RETCODE_INVALID_VOLUME</code>	10016	Khối lượng không hợp lệ
<code>mt5.TRADE_RETCODE_INVALID_PRICE</code>	10017	Giá không hợp lệ
<code>mt5.TRADE_RETCODE_INVALID_STOPS</code>	10018	SL/TP không hợp lệ
<code>mt5.TRADE_RETCODE_MARKET_CLOSED</code>	10024	Thị trường đóng cửa
<code>mt5.TRADE_RETCODE_NO_CONNECTION</code>	10027	Mất kết nối
<code>mt5.TRADE_RETCODE_PRICE_CHANGED</code>	10030	Giá thay đổi
<code>mt5.TRADE_RETCODE_PRICE_OFF</code>	10031	Không có báo giá tại thời điểm gửi lệnh