

Distributed automation

While automated trading is now relatively commonplace, Rickey Cheung has been one of a few to take the process a step further. In addition to his own proprietary trading, he also redistributes trading signals to automated trading clients around the world. AT talks to Cheung about his methods and technology.

How did you become involved in developing and building trading models?

The original stimulus for trading was my biochemistry professor at university who introduced me to trading soybeans. My interest grew from there and during my subsequent MBA in finance. I started building trading models in 2003 on the advice of Dr. Brett Steenbarger, so it was possible to test and verify my strategies.

“...it is easy to fall into the trap of trivialising the time and effort required to build robust trading models.”

Rickey Cheung

What do you regard as the major challenge of building trading models?

Even for people who would classify themselves as professional traders, I think it is easy to fall into the trap of trivialising the time and effort required to build robust trading models. Significant amounts of both are needed because original thought and research is required – following the herd is simply not good enough.

Are you a programmer by training/background?

No, so when I first embarked on developing trading models I hired a specialist programmer and a computer science student who acted as my assistant and had the task of explaining the trading requirements to the programmer. Apart from the risk of misunderstandings (my assistant did not have a trading background) we found that the trading model development platform we were using lacked many of the functions necessary for the ideas I wished to explore. This further delayed things, as it was necessary to build custom functions from scratch.

At what point did you start to consider the distribution of trading signals based upon your models?

While it was obvious that my original trading model was viable by 2002, it was not until early 2004 that I submitted it to Futures Truth for independent testing and verification, with the first results being published in autumn of that year.

How have your models ranked with Futures Truth?

The original trading system (called RC Success) was ranked number one by Futures Truth for more than eighteen months and is still in the top ten. Subsequent models have actually been better performers.

Presumably you have had offers to purchase the source code to your trading models?

Yes, some of them very substantial, but I have preferred to stick with the idea of leasing the output from the models.

How do you distribute this output?

To start with we would supply clients with an encrypted add-in that contain the trading model's logic. Unfortunately it took a while before it became apparent that the encryption used by the

development platform I was using was compromised. Hackers were able, with little difficulty, to crack the encryption and extract the model's rules.

This issue of intellectual property theft is endemic with regard to trading models, which is why security is so critical to prevent information leakage. After this initial experience we tried using a hardware encryption lock, which at first glance looked promising. However, on closer inspection it was clearly not foolproof and was at best perhaps 95% secure. Though it would certainly slow hackers, they would still be able to crack it eventually, which was obviously not good enough for our purposes.

If one is looking to sell the output from a trading model, absolute security of the source code is essential. Otherwise prospective clients have very little incentive to pay a reasonable commercial price for the signals and those who have already subscribed are inevitably aggrieved. As a result, I was obliged to suspend sales of the automated trading signals.

So how did you get around this problem?

Initially, I invested quite a bit of time and capital in trying to develop a programme that would route the orders direct to client accounts, but this didn't work out. However, about three years ago I came across StrategyRunner, which solved all my security concerns at a stroke.

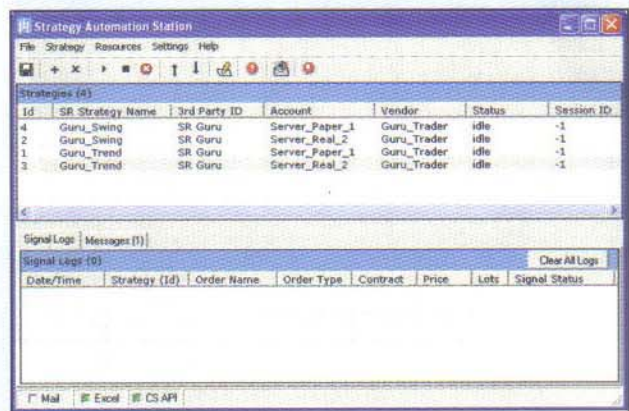


Figure 1: StrategyRunner's Strategy Automation Station

How?

The trading model source code remains entirely within my development platform, which interfaces with StrategyRunner's Strategy Automation Station (SAS). SAS takes the trading signal output from the development platform and routes it to the StrategyRunner Server that is running at each client's FCM's site. The Server then executes the trades on the clients' accounts automatically. →

How much integration work was required on your part to get this up and running?

Effectively none; SAS works with a number of development platforms and it was simply a case of installing the software. The entire process took a matter of minutes before everything was fully operational.

Can the clients monitor the trading process?

They can do more than that. If they run the StrategyRunner Pro Console application they can set up defaults for the number of contracts to trade, stop losses, exits etc. They can then change these in real time if they wish - so if they have a dynamic money management strategy they wish to use, they can apply this to the raw trading signals. They can also choose to ignore signals if they wish or manually confirm them.



"If one is looking to sell the output from a trading model, absolute security of the source code is essential."

How has the order routing performed in terms of reliability?

No problems; it has been completely reliable. The only thing I would say is that because of the interface between SAS and our development platform it is advisable to check any changes before going into production.

By changes, do you mean changes in trading model logic?

No – changes such as updates or patches to the model development platform or upgrades to the machine on which it is running.



Figure 2: StrategyRunner Pro Console - used by Cheung's clients to monitor/control trades triggered by his models

Did you consider any alternative solutions to StrategyRunner?

At the time we couldn't (and still can't) find any alternatives that were acceptable. We have subsequently been approached by a vendor keen to promote and lease the trading models, but one of their requirements was that we re-coded the models in C++. Given that the existing process is running smoothly, the additional effort required to do this didn't seem worthwhile. We have simply found that using StrategyRunner has been reliable and secure.

Which markets do your models cover?

They are specifically intended for the e-mini S&P future, but the same basic principles (obviously with some adjustment for differing volatility levels) are also applicable to the Dow, NASDAQ or Russell e-mini - or just about any US equity index future.

What about timeframe and trade frequency?

The models are intended just for intraday trading so there is no overnight risk. On average they trade once per day, though for the more aggressive models this can be higher. If the signal is not stopped out, the trades will often run for much of the rest of the trading session and exit on the close.

How fast is the trading cycle? Is there much latency?

The complete cycle from the initial signal being generated to clients receiving trade confirmations from their brokers takes a maximum of around three seconds. I monitor that by discussing this regularly with clients to check on performance. I think a significant portion of the trade cycle is probably accounted for by internal application latency on the development platform.

By today's latency standards three seconds seems quite slow?

I think if the models were generating thousands of orders per day, each of which was trying to capture one tick, then this might be an issue. However, the important point is that the models typically only trade once a day, so any latency has a minimal effect. Furthermore, the models usually generate signals before a move develops, so it is not as if they were trying to place orders in the middle of a scramble by the rest of the market to climb aboard.

Are your clients mostly retail traders?

No – absolutely not. There are a very few small retail clients who receive trade signals online. They will typically be trading between one and ten lots per trading signal and can choose from two trading models. However, the bulk of my clients are professional traders – either hedge funds or substantial prop traders who trade full time for a living.

All the professional clients receive the trade signals via StrategyRunner (we insist that they use it). They may be trading a hundred lots at a time, and can choose from among five trading models. In terms of distribution we have clients in Asia, Europe and the US.

Do you have many commodity trading advisers as clients?

No, but interestingly we do have hedge fund clients. In fact one of our newest clients is a US hedge fund that originally approached us and made an offer to purchase the trading logic for one of the models. We were not prepared to do this but the hedge fund was happy to lease the trading signals anyway. At present they are only trading about sixty lots, but they are about to increase this to somewhere between three and five hundred contracts.

Given the number of professional clients you have trading in significant size, do you not have a problem with performance deterioration?

No. Even allowing for the scale of their operations, the total volume of contracts traded is relatively small given the extensive liquidity of the S&P e-mini. There certainly hasn't been any performance deterioration.

