2

Development and Analysis

This chapter is an overview of the process I recommend for development and analysis of trading systems. It is independent of any particular method of trading. It is the method used to develop and analyze mean reversion systems described in this book.

Technical Analysis

Technical analysis and quantitative analysis are based on the belief that several conditions are true.

1. The price and volume reflect all available and necessary information about the company, fund, or market.
2. There are patterns in the records of price and volume that regularly precede profitable opportunities.
3. We can discover those patterns.
4. Those patterns will continue to exist long enough for us to trade them profitably.
5. The markets we model are sufficiently inefficient for us to make a profit trading them.

Technical analysis began as chart analysis, and has developed a large body of subjective interpretation of chart artifacts such as flags, retracements, head-and-shoulders, and trend lines, to name just a few. We traders are all very good at selective vision–we see what we want to see. We can look at a chart and see examples of a big gain following, say, the breakout of a triangle pattern. Thinking we have found a good trade entry technique, we can define those conditions in very precise terms and have the unbiased computer search all the data for instances of that pattern. The results do indeed show a profit for the pattern we saw so clearly. Often it also shows losses from many similar patterns that we either did not see or chose not to acknowledge; and it sometimes shows signals that appear, and then disappear as additional data points are added to the chart.

Quantitative analysis

Quantitative analysis refines technical analysis by:

- Removing the judgment associated with ambiguous chart patterns.
- Defining unambiguous, mathematically precise indicators.
- Requiring that no indicator or signal may change in response to data that is received after it has been initially computed.
- Making extensive use of mathematical models, numerical methods, and computer simulations.
- Applying statistical validation techniques to the resulting trading models.

The mean reversion systems developed in this book are based on the principles of quantitative analysis.

The Signal and the Noise

A trading system is defined to be a combination of a model and one or more series of data. The model contains the rules, logic, and parameters. There is at least one data series—the issue being traded. In many systems, that is both the series examined to detect patterns and the series that is traded. In some systems, auxiliary data series are used to aid in recognition of good entries and exits, or to help define conditions when trades should not be taken.

The data analyzed consists of signals and noise. The signal is that component that contains the patterns that are recognized by the logic of the model; everything not specifically treated as signal is noise, even if there are profitable trading opportunities in the noise. From the point of view of trend following with holding periods of months or longer, swing trades are within what it considers to be the noise. Similarly, day traders and scalpers find profitable trades within what swing traders consider to be the noise. This book focuses on swing trades.

The Development Process

All traders need confidence that their system will be profitable when traded. For any trader, the way to build confidence is to practice.

One advantage that traders who use mechanical systems have over those who use graphical methods is that the mechanical system is (or, at least, can be) objective, rule-based, judgement free, and testable.

You will be writing a trading system, testing it using historical data, then following the rules to place actual trades. The period of time, and the data associated with that time, that you use to develop and test your system is called the in-sample period and the in-sample data. The period of time, and the data associated with that time, that follows the in-sample period and has never been tested or evaluated by the system is called the out-of-sample data. Actual trades are always out-of-sample.

The process of validating a trading system is one of observing the profitability and behavior of the system in the out-of-sample period after it leaves the in-sample period.

The transition you make going from testing your system in-sample to trading it out-of-sample is one data point in the validation of your system. Your confidence level will be much higher if you have observed many of these transitions. The walk forward process automates these practice steps.

Risk of drawdown, profit potential, and position size are all determined by analysis of the out-of-sample trade results.

The outline presented here is designed to build your confidence that your system will be profitable. For a more extensive discussion of trading system development, including expansion of the topics presented in this outline, please see *Quantitative Trading Systems* and *Modeling Trading System Performance*.

1. Objective function

An objective function is a metric of your own choosing that you use to rank the relative performance of two or more alternative trading systems. The phrase "of your own choosing" is critical. Your objective function must accurately reflect what is important to you. If you prefer long holding periods and infrequent trading, your objective function must rate systems that hold for weeks to months higher than systems that hold for a few days. If you prefer high equity gain without regard to drawdown over lower, but smoother, gain your objective function must reflect that.

I believe the psychology of trading experts who try to help traders become comfortable with the systems they trade have it backwards. If you decide ahead of time what you want, and design trading systems that satisfy your wishes, and if you have the confidence built through the validation process, you are guaranteed to be comfortable with your system.

Trading system development platforms, including AmiBroker, report the score for many metrics with each test run. If one of these standard, built-in metrics meets your needs, use it as your objective function. If you want something else, you can create a custom metric and use it. For many people, rewarding equity growth while penalizing drawdown is important. If you agree, consider using one of the standard metrics that does that:

- RAR/MDD (risk adjusted annual rate of return / maximum drawdown).
- CAR/MDD (compound annual rate of return / maximum drawdown).
- K-ratio.
- Ulcer Performance Index.
- RRR (risk-reward ratio).

2. What to trade

This book focuses on highly liquid indexes and ETFs, such as SPY, QQQ, and IWM. Much of the analysis is done using daily data. Some systems compute signals at the close of trading for execution at the close of that bar, some for execution the next day, either at the open or at a pre-computed intra-day price.

3. Design the system

A trading system is a combination of a model and one or more sets of data. The model is contained in the code you write. The data is the price data of the ticker symbol your code processes. The model contains the intelligence. It is looking for the patterns in the data and testing the profitability of buying and selling.

A model consists of several parts: filter or setup, entry, one or more exits, position size, portfolio composition, and so forth. In much of the literature, the entry is emphasized. But the other components are equally important.

A model is a static representation of a dynamic process. Once you are done coding and testing, the model does not change. It may be cleverly designed and have self-adapting parameters, but it is still static. The market being modeled is dynamic and ever changing. Your model is looking for a particular pattern or set of conditions, after which it expects a profitable trade. As long as the model and the market remain in synchronization, the system will be profitable. When the two fall out of sync, the system will be less profitable or unprofitable.

It is useful to think of the data as comprised of two components—signal and noise. The pattern your model is looking for is the signal portion of the data. Everything your model does not recognize is the noise portion of the data. Your goal, as a designer of trading systems, is to accurately recognize the signal and ignore the noise.

Our hope is that:

- We can build a model,
- That recognizes some inefficiency,
- And use that model to trade profitably,
- As long as the model and reality stay in sync.

4. In-sample period

There are two views about how much time and data should be used to develop the system.

Some want a long time. They feel that the system will experience a variety of conditions and be better able to handle changes in the future. The risk is that conditions vary so much during the in-sample period that the system will not learn any of it well.

Some want a short time. They feel that the system will be better able to synchronize itself and learn very well. The risk is that the system may learn a temporary pattern that does not persist beyond the in-sample period. Some of that closer fit is to the noise. A system that fits the noise and performs well in-sample, but does not perform well out-of-sample, is described as a system that is curve-fit, or over-fit, to the data.

The proper length of the in-sample period is impossible to state in general. It is very much a function of both the model and the data. My view is that the length of the in-sample period should be as short as is practical. The only way to determine the length of the in-sample period is to run some tests.

5. Out-of-sample period

The length of the out-of-sample period is: As long as the model and the market remain in sync and the system remains profitable. There is no general relationship between the length of the out-of-sample period and the length of the in-sample period.

6. Optimize

To optimize means to search a lot of alternatives and choose the best of them. Optimization is not a bad thing; in fact, it is a necessary step in system development. If you were to think up a new trading system and write the code for it, it would include logic and parameters that were first guesses. You could test that system as it stands, and then either trade it or erase it and start over. But you are unlikely to do that. More likely, you will try other logic and other parameters values. If you are trying a few hand-picked alternatives, you might as well run an optimization and try thousands of alternatives. Give yourself a chance to find the best system.

Anything you would consider changing in your system—a different parameter value or an alternative logic statement—is a candidate to be optimized. The value of optimization is not so much in searching thousands of alternatives—it is in ranking them and choosing the best. The intelligence is in the objective function.

7. Walk forward runs

While there is nothing special about optimizing, there is something very special about a walk forward run.

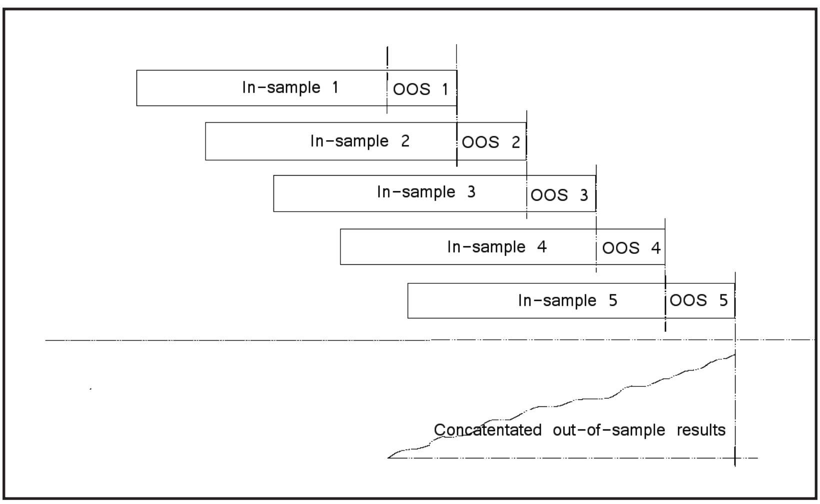

Figure 2.1 -- Walk Forward Part 1

The walk forward process is several iterations of:

1. Optimize in-sample.
2. Choose the best according to your objective function.

3. Use those values and test out-of-sample.
4. Step forward by the length of the out-of-sample period.

Continue this process until you have used the last full in-sample period.

8. Best estimate set of trades

There are two sets of trades produced by the walk forward process—those from all of the in-sample periods, and those from all of the out-of-sample periods. The out-of-sample trades are the reference set. They are the *best estimate* of future performance of the system.

9. Evaluate out-of-sample results

Analyze the best estimate set of trades. Either send the system on to analysis and potentially trade the system or send it back to development.

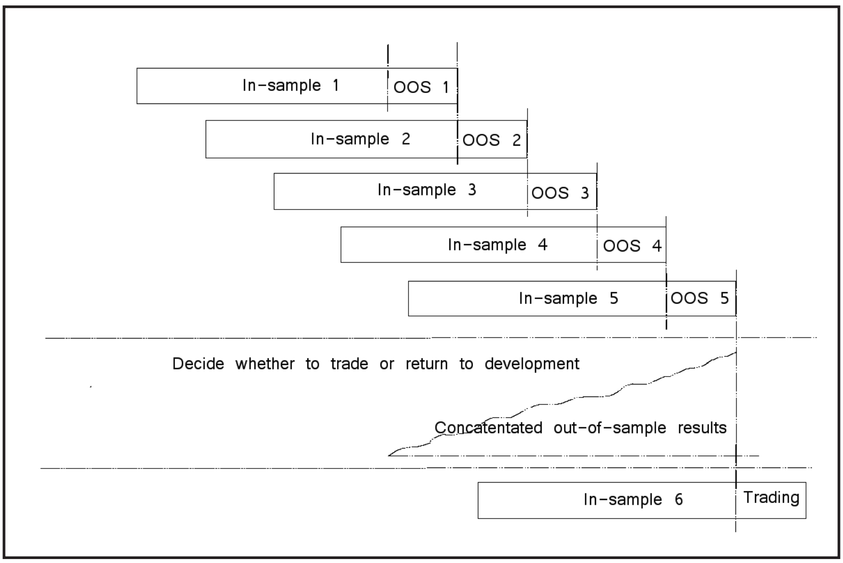

Figure 2.2 -- Walk Forward Part 2

Note the importance of having an objective function you trust. The set of parameters used to test out-of-sample are the parameters that are at the top of the results list, after the list has been sorted by your objective function. The process is automatic and objective. All the decisions were made in earlier steps. You will never even see what the second choice is; the first choice is always used.

Each walk forward step is one data point you will use in the validation of your system.

10. Determine position size

Proper position size is required to meet two goals:

- Maximize account growth.
- Hold drawdown to an acceptable level.

Using the best estimate set of trades, along with your personal and subjective risk tolerance, and the Monte Carlo simulation techniques described in *Modeling Trading System Performance*, determine the position size.

Please note an important point. The system processes the data by applying the rules to generate the entry and exit signals—the system produces the trades. But the system does not determine position size. Position size is a function of trading performance, of the synchronization between the data and the logic, and of the system health. Position size is determined independently of the trading system.

11. Trade the system

Using the set of parameters that are at the top of the list after the last optimization, buy and sell when the system gives signals.

Take all the signals. If you have some way to decide which signals to take and which to skip, that logic belongs in the trading system and should go through the validation process.

On the last day of what would have been the out-of-sample period, re-optimize. Pick the top-ranked parameter values and continue to trade.

12. Monitor the system health

Each trade signaled after development has finished, and each trade you actually take, is an out-of-sample trade. Compare the statistics for your trades with statistics for the out-of-sample results from the walk forward runs.

If your results drop below what is statistically expected, reduce your position size, possibly stop trading the system. Either paper-trade it to see if it recovers; or re-optimize ahead of schedule, then paper-trade it and observe its performance.

Refer to *Modeling Trading System Performance* for detailed discussion of determining whether the system is working or broken, and for techniques for computing the best position size in order to maximize account growth while holding drawdown to a level within the tolerance of the trader.

13. Resynchronize

As trades are taken, modify the best estimate set. Either add the new trades, or replace the oldest trades with the newest trades. As the distribution of those trades changes, the correct position size changes. When the performance deteriorates, as detected by either statistical tests or by the recommended position size becoming smaller, resynchronize the model to the data by repeating the validation steps.

Summary

There are no guarantees. The best we can hope for is a high level of understanding and confidence gained through the validation process.

Distributions

Our goal in system development is to combine a set of logic with a data stream in such a way that profitable trading opportunities can be identified. Short of knowledge of the future, the best estimates of future performance are obtained when two conditions are both true.

- Tests of the system are run using data that has not been used in the development of the system. That is, out-of-sample data and out-of-sample results.
- The data used is representative of the data that will be analyzed and traded.

Emanuel Derman, in his book *Models.Behaving.Badly*, states that models are simplifications; and simplifications can be dangerous. System developers should avoid simplification of data representation. Whenever possible, use distributions rather than a limited number of scalar values.

The information content that describes a trading system over a given period of time can be described in many ways. The following list is in decreasing order of information.

- Reality. Trades, in sequence, that actually result from applying the system.
- List of trades, in time sequence.
- List of trades.
- Distribution of trades.
- Four moments describing the distribution.
- Mean and standard deviation.
- Mean.
- Direction.

Probability and statistics distinguish between population and sample. The population is all items of the type being analyzed; the sample is a subset of the population that has been observed. The purpose of developing trading systems is to learn as much as possible about the population of trades that will occur in the future and make estimates of future performance. The results of testing trading systems form the sample that will be used to make those estimates.

Reality

Reality cannot be known in advance. Estimating reality—the population—is the purpose of system validation. Reality is the logic of the system processing the future data series.

List of trades, in time sequence

The list of trades, in time sequence, that results from processing a set of that data that is similar to the future data is the best estimate we can obtain of reality. There is one of these sequences for each unique set of test data and each set of logic and parameter values. Using these results to estimate future profitability and risk depends on the degree of similarity between the test data and the future data.

List of trades

The list of trades, ignoring time sequence, relaxes the assumption of the trades occurring in a particular sequence. It provides a set of data with, hopefully, the same characteristics as the future data, such as amount won or lost per trade, holding period, intra-trade drawdown, and frequency of trading. Selecting trades from this list in random order gives opportunity to evaluate the effects of similar conditions, but in different time sequence.

Distributions

A distribution can be formed using any of the metrics of the individual trades. The distribution is a further simplification since there are fewer (or at most the same number of) categories for the distribution than for the data used to form it. For example, a distribution of percentage gain per trade is formed by sorting the individual trades according to gain per trade, establishing ranges and bins, assigning each trade to a bin, and counting the number of trades in each bin. A plot of the count per bin versus gain per bin gives a plot of the probability mass function (often called the probability density function, pdf).

Four moments

Distributions can be described by their moments. The four moments most commonly used are mean, variance, skewness, and kurtosis. Depending on the distribution, some of the moments may be undefined.

- Mean. The first moment. The arithmetic average of the data points.
- Variance. Second moment. A measure of the deviation of data points from the mean. Standard deviation is the positive square root of variance.
- Skewness. Third moment. A measure of the lopsidedness of the distribution.
- Kurtosis. Fourth moment. A measure of the peakedness and tail weight of the distribution.

Mean and standard deviation

Mean and standard deviation are commonly computed and used to describe trade results. They can be used in the definition of metrics such Bollinger Bands, z-score, Sharpe ratio, mean-variance portfolio, etc.

Mean

The mean gives the average of the values. Mean can be computed in several ways, such as arithmetic mean and geometric mean. The mean is the single-valued metric most often associated with a set of data.

Direction

Direction of a trade describes whether it was a winning trade or a losing trade. Direction is meant to represent any way of describing the trades in a binary fashion. Other ways might be whether the result was large or small in absolute value, or whether the maximum favorable excursion met some criterion, etc.

Stay high on the list

With each step down this list, a larger number of data points are consolidated into a smaller number of categories, and information is irretrievably lost. Knowing only the information available at one level makes it impossible to know anything definite about the population that could be determined at a higher level. Working with only the mean tells us nothing about variability. Working with only mean and standard deviation tells us nothing about the heaviness of the tails. Using the four values of the first four moments enables us to calculate some information about the shape of the population, but nothing about the lumpiness or gaps that may exist.

Application to modeling trading systems

The Monte Carlo simulation technique repeatedly builds and analyzes a sequence of trades. Each sequence represents a fixed time horizon, say two years. The number of trades in that sequence is the number of trades that can be expected to occur in two years. Each trade is selected from either a trade list or a distribution that represents a trade list. When enough trades to cover two years have been selected, the equity curve, win-to-loss ratio, maximum drawdown, and other metrics of interest can be calculated–just as if this sequence was a two-year trade history.

After generating and analyzing many, usually thousands, of two-year sequences, form and analyze the distributions of the results. Two of the most important metrics of results are account growth and maximum closed trade drawdown. Others that might be of interest include win-to-loss ratio, maximum intra-trade drawdown, etc.

The important points are:

- Use the best data available. Out-of-sample data that best represents the anticipated future data that will be traded.
- Use the highest representation of the data possible. That is, either a list of trades or a detailed distribution of trades.
- Using simulation techniques, generate many possible trade sequences.
- Analyze the distributions of the results of the simulation.
- Calculate and use metrics based on the distributions to judge the usefulness of the system.

Further reading

There are several books listed on the Resources page of my website that have excellent discussion of the importance of distributions and the dangers of using only single values. If you can read just one, begin with Sam Savage, *The Flaw of Averages: Why We Underestimate Risk in the Face of Uncertainty*.

The Importance of Consistency

Backtest results depend on the fit between the model and the data. As developers, our first goal is backtests that are highly profitable with smooth equity curves. We hope these results are indicative that the logic is accurately identifying profitable patterns—distinguishing the signal from the noise. But until the system has been validated we cannot tell what portions of the results are simply data mining.

The validation process, particularly the walk forward process, provides us with a set of trades that are the best available estimate of how the system would perform as it is periodically resynchronized and traded. Even at that, some of the out-of-sample results were detected as patterns, but are random values that just happen to fit the rules; there is no profitable trade following them. Further, the walk forward process provides a single sequence of trades. If the future data is identical to the past data, that sequence might be repeated. If the future is similar (has the same distribution of trades), but trades occur in a different order, the equity curve will have a different shape.

Figure 2.3 shows ten equity curves, each created by randomly selecting 50 trades (the average number of trades in a two year period) from the best estimate set associated with the naive system described in Chapter 3. Each trade is made with all available funds. The average return per trade is 0.41%. The average equity after 50 trades is expected to be $122,701 (100000*1.0041^50). The average of the ten on the chart is the dark dashed line, with a terminal equity of $127,650. If you were trading the system for two years, your results could be curve A, or B, or C—all with equal probability.

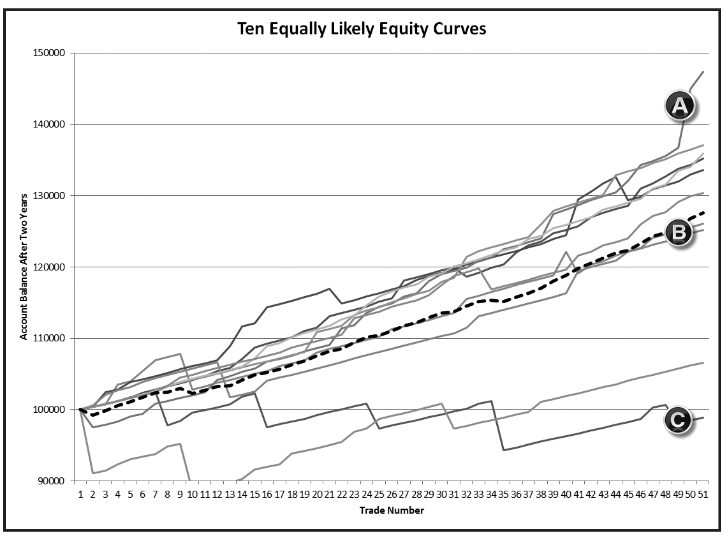

Figure 2.3 - Ten equally likely equity curves

The consistency of the trades determines the predictability of the account equity. That, in turn, determines risk, position size, and profit potential.

As market conditions change, and trades taken by the system are added to the best estimate set, the distribution of future equity curves changes accordingly. Those actual trades are important for monitoring system health and dynamically adjusting position size.

Risk

Risk, as I define it here, is measured as percentage drawdown from the highest balance of the trading account. Risk, and profit potential, begins with the individual trade, then progresses to the sequence of trades.

Each trader must understand the characteristics of the system being traded. As related to risk, intra-trade adverse excursion is important.

Your system has exit rules that will eventually generate a signal to close every trade. If you are always willing to hold positions until the system gives the signal, then the drawdown is based on the profit or loss of closed trades. If you would override your system and close a trade because it had a large intra-trade loss, then the drawdown is based on intra-trade profit or loss.

I understand that there will be unusual and unexpected conditions. But I recommend that as many of those as possible be anticipated, code to generate the appropriate signals be added to the system logic, and the effect tested. The code that implements what would be an override of the logic is a maximum loss exit.

The extend to which the intra-trade drawdown is important depends on the goal of the system. A swing trading system expects to hold a short period of time, is often taking positions that are oversold, and regularly has an intra-trade loss at some point during the trade. A trend following system expects to hold a long period of time. While it is waiting for the large profit trade, it is exposed to larger intra-trade drawdowns. Swing traders are less likely to override their systems due to an intra-trade drawdown, and, consequently, closed trade results are appropriate for analyzing risk.

As I discuss later in this chapter, the primary reason traders stop trading a system is a drawdown in account equity that exceeds their tolerance.

Risk tolerance is personal and subjective. It is expressed as a combination of two numbers.

One is the maximum drawdown the trader is willing to accept before taking the system offline. This might be 10, 15, or 20%. If a trading account had grown to a balance of $200,000, a drawdown of 10% is a drawdown of $20,000. Drawdowns always increase over time. If a system is likely to have a 10% drawdown over a one year horizon, over a two year horizon under the same circumstances it will have about a 14% drawdown. Drawdown increases in proportion to the square root of holding period for individual trades, and in proportion to the square root of the forecast period for system performance estimates.

The other is the confidence the trader wants about the drawdown limit. That is expressed in terms of the percentile of the distribution of drawdowns. A desire to be 90% confident means that if the system trades for ten two year periods, the maximum drawdown will not exceed 10% in nine of those periods.

Every trading system has some probability of experiencing a sequence of losing trades that exceed the trader's tolerance. The best estimate set of trades, together with the Monte Carlo simulation described in *Modeling Trading System Performance*, can be used to quantify that probability.

Position Size

Once the risk is quantified, the maximum position size—the maximum fraction of the account used for each trade—that can be used without the drawdown exceeding a tolerance limit can be computed. The position size is independent of the trading system itself. It can be computed from the size of the trading account, the best estimate set of trades, and the trader's risk tolerance.

Potential Profit

Given the maximum position size, the distribution of account equity can be computed and translated back to compound annual rate of return, CAR. The metric of CAR / Maximum Drawdown is useful in deciding whether a system is worth trading.

System Health

System performance and system health fluctuates as the data changes and the degree of synchronization between the model and the data changes. Account growth and drawdown change accordingly.

Dynamic

Every system is dynamic. As trades are made, those trades should be added to the best estimate set, and the updated set used to reassess risk and adjust position size.

Schematic of a Trading System

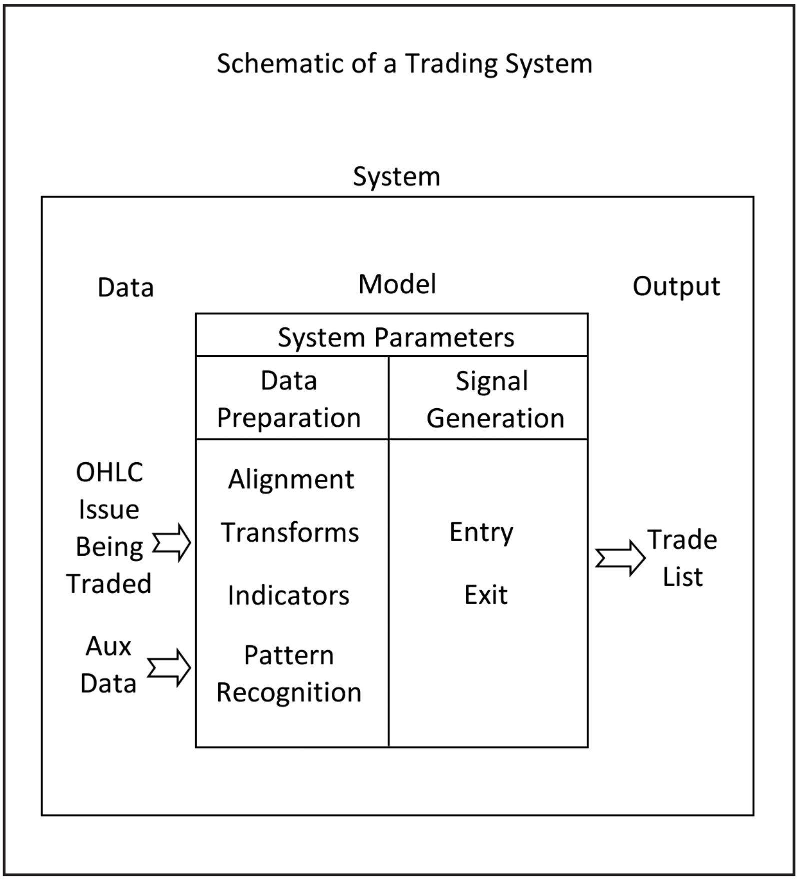

Figure 2.4 -- Schematic of a Trading System

Flowchart of System Development and Trading Management

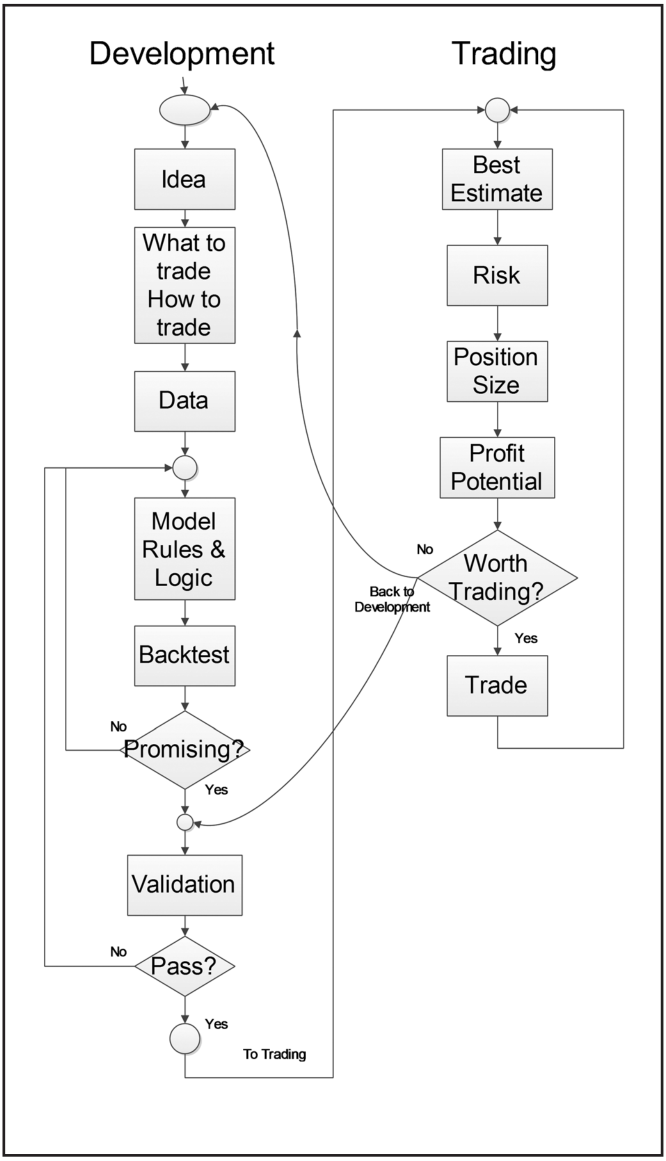

Figure 2.5 -- Flowchart of System Development and Trading Management

Dissonance Alert

Expert advice says both:

- Ignore the in-sample results!
- Have confidence in the in-sample results!

How can both be reasonable?

During the early phases of development, the logic and parameters are adjusted to give the best fit to the in-sample data. If that fit is to the signal portion of the data, and the characteristics of the data remain consistent in the following out-of-sample data, then the out-of-sample performance will be good.

The only way to fit the model to the data is through in-sample optimization. The only way to verify that the fit is profitable is by testing out-of-sample data.

In the final analysis, the system relies on good in-sample results to select the rules, determine the parameters, and identify the signals. But you cannot have confidence that the system is reliable without good out-of-sample results.

Managing Subjectivity with Objective Functions

Every day, traders must make decisions:

- What to trade.
- When to enter.
- How large a position to take.
- When to exit.
- Whether the system is healthy.

Discretionary traders acknowledge the subjectivity of their decisions and draw on experience. Systematic traders use objective functions designed to identify important decision criteria and quantify them.

Objective functions are important in two phases of trading systems design:

- To rank alternative systems during system development.
- To decide the size of the next position during system analysis.

Ranking alternative systems

A trading system is a set of computations, logic statements, and parameter values that comprise a set a rules that identify profitable trading patterns and give buy and sell signals.

There are an infinite number of possible systems. In order to make the process manageable, relatively simple systems are designed to focus on specific trading ideas, such as trend following, mean reverting, seasonality, etc.

For any one of these ideas, there are many alternatives. A trend following system might have logic that looks at breakouts, the crossing of two moving averages, or the projection of a regression. For each of these there are numeric parameters such as the lengths of the moving averages, or magnitude of breakout. There might be multiple rules to exit a position, such as logic, trailing exit, profit target, and / or maximum loss stop.

Designing a trading system is an iterative process of:

- Modify the logic and parameters.
- Test the performance.

Each set of logic and parameters creates a new trading system – one of the alternatives to be evaluated. The developer needs to decide which is *best*.

Best is subjective. The purpose of the objective function is to provide an objective metric that represents the subjectivity of the developer's definition of best.

Depending on the preference of the trading system developer, determination of best includes evaluation of profitability, risk, trading frequency, holding period, execution efficiency, percentage of winning trades, and any other measurable feature.

A decade ago, the reporting from test runs made by trading system development platforms was limited to basic metrics such as net profit and maximum drawdown. Today, many platforms provide both a wide range of sophisticated metrics and the capability for the developer to design custom metrics. This gives the developer the opportunity to incorporate subjective judgments into a custom objective function–a metric computed using scores and weights for each of the evaluation criteria.

To insure that the objective function accurately reflects the preferences of the developers, it must be calibrated. The calibration process is:

1. Pick a time period and tradable issue and apply a trading system, generating a set of trade results and equity curve.
2. Perform Step 1 several times, using different logic and different parameter values. Using the same time period and the same issue, generate the reports and equity curve for each alternative.
3. Print out the reports and equity curves, one to a page, and lay them out on a table or floor.
4. Arrange them into order according to your subjective preference—from the one you would be most comfortable trading to the least. If necessary, make up and include hypothetical results to fill in gaps in performance and to represent results typical for your trading.
5. Analyze the results, paying particular attention to the features that are important to you. Create a list of these features. Create a measurement scale for each of the features, and assign relative weights to each feature. Set up the measurement scale so higher values are preferred. The sum of all the measurement scores and weights is your objective function.
6. Return to the trade reports and calculate the objective function score for each report. This step should be very easy. If it is not, return to Step 5.
7. If the order the reports were placed in using your subjective judgment is the same as the order of the objective function, you have created an objective function that reflects your subjective preferences. You are done.
8. If the two orderings are different, modify your objective function

by adding or removing terms and modifying weights. Return to Steps 5, 6, and 7 until you are satisfied.

When you are satisfied, program your objective function and use it to objectively rank alternatives. Whenever you find that you prefer some alternative other than the one ranked highest, return to the design of your objective function and refine it. The goal is to have a high degree of confidence that trading results that have high objective function scores are results you like, and the systems that produced them are systems you would be confident trading.

I favor functions that reward account growth, reward smoothness, and penalize drawdown. Several well documented functions that include these criteria are:

- RAR/MDD (risk adjusted annual rate of return / maximum drawdown).
- CAR/MDD (compound annual rate of return / maximum drawdown).
- K-ratio.
- Ulcer Performance Index.
- RRR (risk-reward ratio).

Upon completion of calibration, the designer can be confident that the trading system ranked highest among all the alternatives is the one he prefers. This is useful during the design and testing phases. It is critically important during the walk forward testing in the validation phase because the model with the highest objective function score is automatically used to trade the out-of-sample period.

For more discussion about objective functions, see *Quantitative Trading Systems*.

Deciding the size of the next position

The trading system that results from the design, testing, and validation provides a single set of trades with single mean, single standard deviation, single terminal wealth, single maximum drawdown.

These results will be repeated as the system is traded only if the future price series is exactly the same as the historical series used during development. In order to estimate profit potential and risk it is important to consider the distribution of potential results.

Modeling future performance, including evaluating system health, estimating profit potential and estimating risk is based on:

1. Using the set of trade results that, in the judgment of the developer, best represents the trades that are likely to occur in the future.
2. Using Monte Carlo simulation techniques to create equally likely trade sequences.
3. Analyzing the distributions of profit and drawdown resulting from the trade sequences.
4. Comparing both the magnitude and probability of both the profit potential and drawdown with the trader's personal tolerance for risk and desire for profit to determine system health and position size.

An objective function formed by taking the ratio of the Compound Annual Rate of Return (CAR) at some confidence level to the Maximum Drawdown (MaxDD) at some confidence level is very useful in deciding whether a system is worth trading.

This process, including the software necessary to run the Monte Carlo simulations, is described in detail in *Modeling Trading System Performance.*

Position size is intimately related to system health and changes dynamically as the synchronization between the logic of the trading system and the price series it is processing changes. System health must be monitored during trading, and position size revised as necessary.

Optimization and the Curse of Dimensionality

On a forum I regularly follow, Yahoo AmiBroker Forum, a message complained that the AmiBroker trading system development platform was unable to handle an optimization that had 10 parameters, each with 20 values. Whether that poster was serious about the numbers 10 and 20 or not, the question is worth considering.

A large number of parameters, say 10, each with a large number of evaluation points, say 20, leads directly to the "curse of dimensionality." Assuming that it takes one nanosecond to evaluate each alternative, and that everything works as it should and the run does finish (no power failure, memory failure, cpu failure, hard disk failure, flood, earthquake, premature death, etc), exhaustive evaluation of the n = 10^20 points will have taken about 3100 years. At 50 lines per 12 inch page, the listing of the results will be 65 light years long. (The nearby universe is described as being those bodies within about 15 light years.)

Each result gives the value of an objective function. Sorting the results into ascending or descending order according to the value of the objective function puts the set of values that are "best" at the top of the list. Assuming a sorting algorithm that takes Order (n*ln(n)) is used, it will take an additional 46 times 3100 years to sort them.

Now make an individual run using each of the sets of values near the top to gain some confidence that the alternative ranked first is the one that is preferred. Maybe an hour or two more?

Just in time to give a signal for tomorrow's trade?

What is a reasonable number of alternatives to evaluate?

I recommend beginning by giving a considerable amount of thought and experimentation to design and testing of the objective function. Have confidence that the alternative with the highest objective function score is the one that is preferred. Or at least that it is acceptable—not letting the search for perfect get in the way of finding a satisfactory solution.

Run some tests using your system, your objective function, your data, on your computer. How long does it take to generate and sort 1000 or 10,000 alternatives? How many of these test runs do you expect to make before deciding on the logic and parameters that will be used? How many hours are you willing to allocate to the process? Working the arithmetic backwards will give an estimate of the number of alternatives that can realistically be tested. Continuing to work backward, you can compute the size of the parameter space that can be exhaustively processed in that amount of time. Make some trial runs using both exhaustive and non-exhaustive methods. Compare the results that rank best from each set of runs. Decide whether you can accept the results of the non-exhaustive method. If so, use it; if not, modify the afl code so that the number of parameters and number of evaluation points for each create an examination space that can be processed in the time you have allocated.

Why Traders Stop Trading

Assume a trader has a method – mechanical, discretionary, or a combination of both – that she has been using successfully. Also assume that she understands both herself and the business of trading, and wants to continue trading. Why would she stop trading that system?

Here are a few possibilities:

1. The results are too good.
2. The results are not worth the effort.
3. The results are not worth the stress.
4. She has enough money.
5. There is a serious drawdown.

1. Results are too good, and she is afraid that this cannot possibly continue. Her system – any system – works when the logic and the data are synchronized. There are many reasons why systems fail and should be taken offline, but a sequence of winning trades should be seen as a success.
   She should continue trading it until one of the other reasons to stop happens.
2. The results are not worth the effort. There is not much gain, but not much loss either. On balance, the time, energy, and resources would be more productively applied doing something else.
   The problem is with the system. She can return the system to development and try to refine it; or take it offline and periodically review its performance.
3. The results are not worth the stress. Performance is satisfactory, but at a high cost – worry and loss of sleep. Regardless of the position size indicated by the distribution of risk, the positions being taken are too large.
   She should either reduce position size or have someone else execute the trades.
4. She has enough money.
   No matter how good a system is, there is always a risk of serious loss. When she has reached her goal, she should retire.
5. There is a serious drawdown. The magnitude of the drawdown needed for it to be classified as serious is subjective. Among my colleagues and clients, those who manage other people's money typically want drawdown limited to single digits. Those trading

their own money may be willing to suffer drawdowns of 15 or 20 percent.

But for every person there is a level of drawdown at which he stops trading the system – preferably while the account still has a positive balance.

My view is that experiencing a large drawdown is the primary reason people stop trading a system.

What causes a large drawdown and how should the trader react to it?

1. The system is broken.
2. There was an unexpected sequence of losing trades.
3. The system is out of synch.
4. The position size is too high.

As the account balance drops from an equity high into a drawdown, it is not possible to determine which is The reason.

All of the reasons are true to some extent. A system that is broken breaks because the logic and the data become unsynchronized, causing an unexpected sequence of losing trades and at a time when position size was too high for conditions.

The solution is two-fold:

1. Continually monitor system performance and system health.
2. Continually modify position size to reflect recent performance.

During the trading system development process, a baseline of system performance is established. The out-of-sample trades from the walk forward phase is a good source of this data. Personal risk tolerance and system risk, taken together, determine position size for that system performance. As system performance changes, position size must also change.

Position size varies in response to system health.

Conclusion

Do not continue to trade a system that has entered a serious drawdown expecting that it will recover. It may recover on its own; it may require readjustment; or it may be permanently broken and never work again.

Take it offline and either observe it until recent paper-trade results demonstrate that it is healthy again, or send it back to development.

The correct position size for a system that is broken is zero.