



This item was submitted to Loughborough's Institutional Repository (<https://dspace.lboro.ac.uk/>) by the author and is made available under the following Creative Commons Licence conditions.

 **creative commons**
C O M M O N S D E E D

Attribution-NonCommercial-NoDerivs 2.5

You are free:

- to copy, distribute, display, and perform the work

Under the following conditions:

 **Attribution.** You must attribute the work in the manner specified by the author or licensor.

 **Noncommercial.** You may not use this work for commercial purposes.

 **No Derivative Works.** You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the [Legal Code \(the full license\)](#).

[Disclaimer](#) 

For the full text of this licence, please go to:
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

Developing Trading Strategies based on Risk-analysis of Stocks

Martin Sykora and Sameer Singh

Research School of Informatics,
Loughborough University, Loughborough, UK
M.D.Sykora@lboro.ac.uk, S.Singh@lboro.ac.uk

Abstract: Risk Management has always been of fundamental importance to financial markets. The aim of all good trading strategies is based around minimising possible risk and at the same time achieving most profit. A balance between these two factors must be struck for different risk – profit profiles. In this paper we describe an innovative way for visually quantifying risk, and we show how our method can be used as a tool for developing trading strategies to help manage risk. We run our algorithm on selected historical FTSE-100 stocks and pick some companies for a more detailed study of trading strategies. The method shows considerable promise for future research work.

9.1 Introduction

For many years now the most widely used method for measuring risk in financial assets has been standard deviation of univariate time-series data [1, 2]. This method and its derivatives are used by traders, brokers and professional fund managers all around the world because of its simplicity and well-known statistical properties. Standard deviation measures the spread of distribution about its mean, and as such does not take into account any of the investors' ability to predict asset price moves. Standard deviation is a very rough measure in the financial context and can be misleading, and there is still disagreement on what constitutes risk and how best to quantify it [2].

We propose a method to quantify risk in a more meaningful way for those investing in stock markets. The core characteristic of our approach is based on the postulation that risk is a function of our ability to forecast an asset, time-horizon over which risk

Martin Sykora and Sameer Singh

is measured and stock data complexity, $Risk = f(\alpha, \beta, \gamma)$, where α is Predictor, β is Time-Horizon and γ is Complexity. To illustrate this let us assume that an investment agent is able to anticipate future moves with good accuracy. Hence the agent would most likely feel confident making an investment. On the other hand if an agent has little idea of whether a market will move up or down it is probably best to stay out of the market. The basic idea that risk is directly related to our prediction ability is intrinsic to many trading decisions. However we are not aware of any meaningful or novel attempts at quantifying risk in this manner.

The rest of this paper is structured as follows; in the next section we discuss our risk quantification method, in section 9.3 we show how we can use the method to select trading strategies, section 9.4 presents results to illustrate the potential benefit of the method and in section 9.5 we present key conclusions of our work.

9.2 Risk Quantification

We propose a risk quantification strategy based on Prediction Engineering and Risk Limitation (PEARL) model. The key elements of our approach involve modeling and visualization of risk based on our ability to predict several time steps ahead because we can develop trading strategies that are based on finding the right time in the future to sell or buy. Our strategy builds a RQG (Risk Quantification Graph) which a trader can visualize or an automated trading software can use directly to invest in the stock markets. A sample RQG is shown in Figure 1. The x -axis of the graph measures the amount of risk involved in trading t time steps ahead, and the y -axis measures the profits or benefits involved with that, provided we use n predictors (e.g. non-linear regressor such as neural network) for a maximum of n time step ahead forecasts. It is important to note that in this approach for a j step ahead prediction, where $j \leq n$, predicted values obtained from all predictors predicting 1 to $j-1$ steps ahead are used as input for the forecast. The maximum time horizon n also determines the duration within which RQG has any value, i.e. the trader should trade within this period to make use of the graph. Our risk quantification method is based on several processing steps which are described in the following sections.

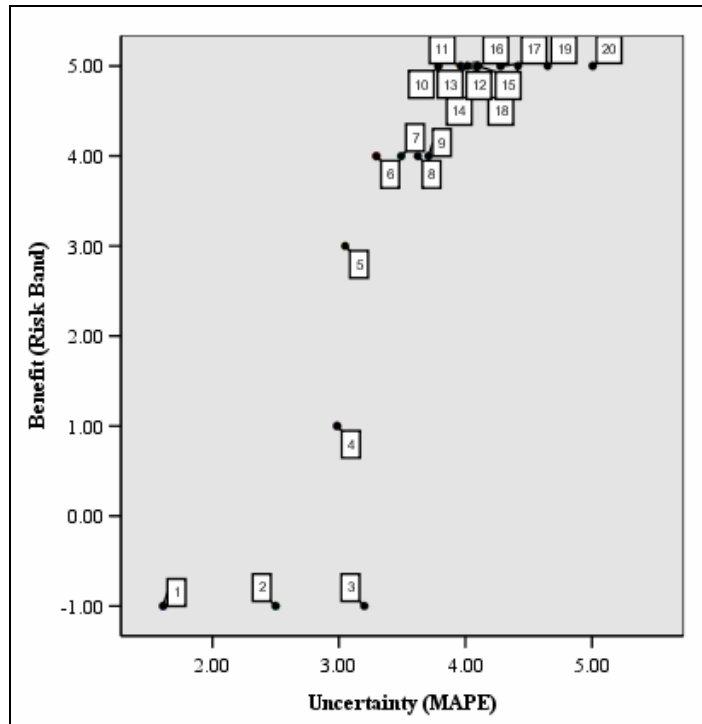


Fig. 1. (Wolseley 8/11/2006 – 5/12/2006) Risk Quantification Graph. 1 – 20 day Ahead predictions and associated uncertainty are plotted above.

9.2.1 Developing Forecast Models

We first divide the data into three parts: training, validation and test set. The first step is to develop a total of n predictors for a given stock. The aim is to use the training data to teach our j th predictor how to perform j time steps ahead forecast, where $(1 \leq j \leq n)$, use the validation set optimize the parameters of the predictor (e.g. in Brown's Double Exponential Smoothing predictor we need to optimize for α), as well as determine uncertainty associated with each of the predictors (this is explained in detail in section 9.2.5), and finally use the test set with our trading strategy as explained in section 9.3 to judge how well we can both predict as well as effectively trade to make profits and minimize risk.

The overall process of developing predictors involves the following steps:

Martin Sykora and Sameer Singh

9.2.2 Choose a predictor

A number of predictors are available that work on univariate time-series data. In this paper such prediction is our immediate concern and the interested reader for multivariate forecasting is referred to [3]. Any predictor has an input and an output. The input data is often the lagged time series values. For example, consider your time series to be $(x_1, x_2, \dots, x_i, \dots, x_N)$ with N data points. To predict data point x_i , we can use the last m time steps, i.e. input data is $(x_{i-1-m}, \dots, x_{i-1})$ and the output is x_i . Since both the input data and output are numeric values, any linear or non-linear regression model can be used. We will call the above approach as Real Value Prediction (RVP). However, recently it has been recognized that for developing and using effective trading strategies it is not necessary to have an accurate forecast of the true stock value. In the very basic form, predicting whether a stock price will go up or not, or the extent to which prices will vary (let us say using fuzzy linguistic variables *very high*, *high*, *low*, *very low*) would be sufficient for making trading decisions. Lindsay et al. [4] have suggested that therefore regression problem can be treated as a classification problem by banding the output values in a total of B bands, and then using the input data of the form $(x_{i-1-m}, \dots, x_{i-1})$ to predict which of the bands (B_1, \dots, B_L) the data point x_i will lie in. The value of L can be optimized on the validation set or can be user specified to maximize the benefits with the selected trading strategy. We call this approach Band Value Prediction (BVP).

In this paper we have selected one example with RVP, using Brown's Double Exponential Model, and another example with BVP using a k -nearest neighbour classifier. Browns Double Exponential Smoothing model [5, 6] model is essentially averaging last m points based on a parameter, α and it also caters for the trend in a series. The method is not able to predict turning points in the series because it only models simple linear relationships, but the method works reasonably well in predicting stable trends [5]. For a more complete description of the model please consult the two sources, referenced above. Due to the stock markets' non-linear nature [7], predictors that model non-linear relationships would likely perform better. The second predictor we have used is a k NN or k nearest neighbor classification algorithm [8]. k NN however needs considerably more data than Brown Double Exponential model.

Since our trading strategy, as well as Risk Quantification Graph is designed to use the predicted value in banded regions (i.e. rather than specifying x_i we specify its corresponding band), we use a banding approach (which is different to BVP) on the output of RVP. Our output banding approaches are explained next.

9.2.3 Data preparation for Real Value Prediction

A given time series $(x_1, x_2, \dots, x_i, \dots, x_N)$ can be predicted either on its actual values or after making it stationary. Difference and log operators are often used to stationarise time series. We recommend that all time series data should be made stationary, especially with respect to the mean before predictions are performed. When we chose to use Brown's Double Exponential Model we found that better results were obtained without making the data stationary, and hence we have decided to use the data as given. The output x_i is now banded as follows. We take the data $(x_{i-1-20} \dots x_{i-1})$, and find its minimum (a) and maximum (b) . A total of 11 bands of size $\left(\frac{b-a}{11}\right)$ are centred around x_{i-1} . The prediction success of the model is based on whether the actual and predicted output value bands are the same or not, or can be based on mean absolute percentage error of predicted returns.

9.2.4 Data preparation for Band Value Prediction

We have used a k-nearest neighbour classifier for band-value prediction. In this specific case we find that differencing the time-series twice and predicting this is better than predicting original value. A first differenced time series is of the form $(x_2 - x_1, x_3 - x_2, \dots, x_i - x_{i-1}, \dots, x_N - x_{N-1})$. Let us denote the final differenced time series to be predicted as $(y_1, y_2, \dots, y_j, \dots, y_{N-2})$. The difference between the minimum and maximum of the training data is given by minimum (a') and maximum (b') . We experiment with $L = 2, 4, 6$ bands and these are centred on the x_{i-1} with a width of $\left(\frac{b'-a'}{L}\right)$. The predictive ability is now measured based on whether the actual and predicted output value bands are the same or not.

9.2.5 Measuring Uncertainty of Forecast Models

The RQG is generated with the basic assumption that forecast ability determines risk. If you have a perfect predictor for the future, there is no risk. Otherwise the level of risk is dependent on the quality of predictor, data complexity, and how far we wish to predict. In fact all of these three factors are correlated. The x-axis of RQG quantifies the risk involved by predicting above 1, 2, ... n steps ahead. Hence uncertainty associated with predicting m steps ahead can be measured as the accuracy on validation set that is achievable with a predictor model trained to predict m steps ahead. To illustrate this, let's say for a j step ahead forecast, on a total of m validation data points, the error will be measured m times as $(e_1^{(j)}, e_2^{(j)}, \dots, e_m^{(j)})$. The average

Martin Sykora and Sameer Singh

error estimate will be $\hat{e}(j) = \sum_{p=1}^m e_p^j$. This is our measure of uncertainty on j step ahead forecast.

9.3 Trading Strategies

After generating the RQG described in previous section we can now attempt to visually interpret it and discuss its role in trading strategies. Figure 1 shows an example graph where the most optimal investment situation is the far top left corner corresponding to the most benefit and least uncertainty. This is an ideal high profit – low risk situation. It is generally believed that financial-agents are risk-averse investors [2], which means that investors prefer more wealth to less but to accept more risk they also require higher expectations of returns. Hence it is of interest to find situations with an appropriate risk – return balance.

In order to rank trading time-horizons on most optimal risk – return we can compute Euclidean distances (Eq. 2) from a given point (x_i, y_i) on the scatter-plot to the most optimal point (left – top corner, coordinate $(0, 5)$). Before we compute the distances we use standard min-max normalisation to transform uncertainty and benefit into $0...1$ ranges. With the Euclidean distances known we can now rank our trading time-horizons based on the minimum distance. Table 1 and Figure 2 illustrates this (do not worry at this point in time how the figure was derived).

$$d = \sqrt{(x_i - 0)^2 + (y_i - 5)^2} \quad (2)$$

Some investors are more risk averse than others this can be because of age, family/cash flow situation and other motivations. To cater for this we can also compute weighted Euclidean distances instead (Eq. 3), where a weight in range of $0...1$ is given to uncertainty (w_1) and a weight in range of $0...1$ is given to the benefit (w_2). This causes the rankings of time horizons to change to new most optimal rankings. For example when a weight of 0.3 is assigned to benefit and weight of 0.7 to uncertainty, this indicates that it is important for us not to take on much risk, and a weight of 0.3 on profit means that our goal are not high returns but returns do carry some importance

$$d = \sqrt{w_1(x_i - 0)^2 + w_2(y_i - 5)^2} \quad (3)$$

Table 2 shows rankings of different weights for Fig. 2, where two sets of weights were used. In the first set of weights, (*uncertainty*) $w_1 = 0.7$ and (*benefit*) $w_2 = 0.3$, in the second set of weights $w_1 = 0.3$ and $w_2 = 0.7$. The row *ranking*, represents the

Chapter 9 - Developing Trading Strategies based on Risk-analysis of Stocks

order of best to worse trading horizons as based on Euclidean distances, where ranking 1 stands for the smallest Euclidean distance. In the first set of weights, the ranking of trading horizons is biased towards minimising uncertainty, and hence position of the points on y-axis plays less of a role. The second set of weights favors profit over uncertainty, hence for example point 3 is ranked before 1, or point 19 before 12.

RQG can be used in various interesting ways to implement real trading strategies. When time-horizons are above the no-profit/loss line, see Fig. 2, then, this indicates that predictions for those investment-horizons are in positive returns. Hence a possible trading strategy would be to buy a stock for all time-horizons above the no-profit/loss line. So essentially we would buy at time t and sell at time $t+h$, where h is a time-horizon above no-profit/loss line, this means we would buy and sell as many times as there are time-horizons above the line $y=0$.

This strategy can very much be reversed, where we sell stocks for all time-horizons below no-profit/loss line and buy them back at their relevant time-horizons. This process is known as *short* selling and is a commonly used trading method [9]. Further we could fine-tune the strategy by only trading time-horizons with low uncertainty, where given some readiness to risk we only consider trades with uncertainty lower then an appropriate threshold. In Fig. 2 we see a vertical line at $x=4$, based on which we filter out all other trades but everything that is to the left of the line.

<i>Time Horizons</i>												
<i>Ranking</i>	1	2	3	4	5	6	7	8	9	10	11	...
	2	3	1	4	5	12	19	18	7	15	8	...
<i>continued...</i>	12	13	14	15	16	17	18	19	20			
	20	6	16	17	10	11	9	13	14			

Table 1. Time-horizon ranking based on Euclidean distances

<i>Weights</i>	<i>Time Horizons</i>															
<i>Ranking</i>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...
U=0.7 B=0.3	2	1	3	4	5	12	7	8	6	10	11	9	19	18	15	...
U=0.3 B=0.7	2	3	1	4	5	19	18	20	12	15	16	17	7	8	6	...

Table 2. Time-horizon ranking (top 15) based on Euclidean distances

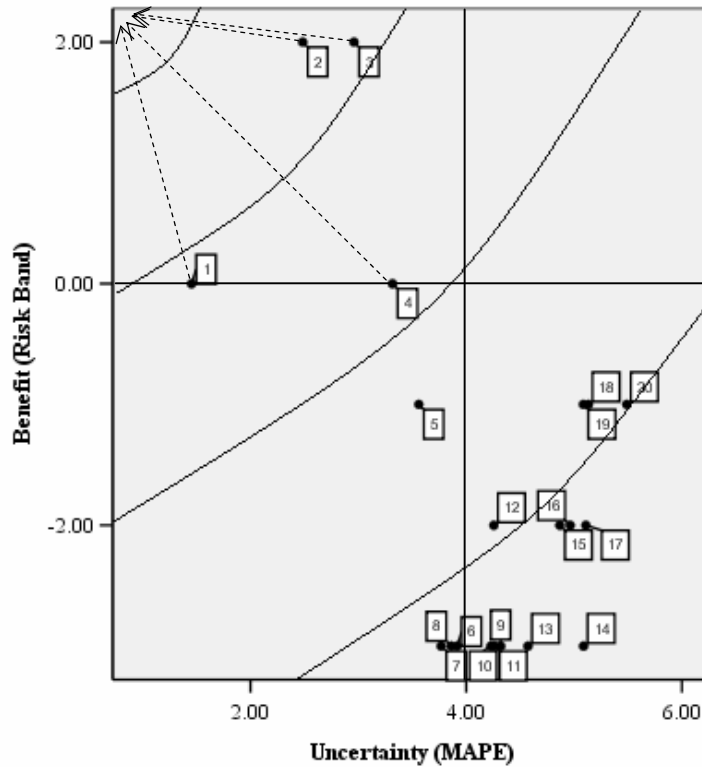


Fig. 2. (AstraZeneca 6/12/2006 – 5/1/2007) RQG, Desirability of trading 1 – 4 steps ahead can be based on Euclidean distance between their position and ideal (0, 2) position shown above

One obvious strategy that can be personalized to an agent's investments needs and requirements follows directly from assigning different weights on uncertainty and risk. Suppose an investment agent has motivation for high profits, then such weights would be set that $w_2 = r$ and $w_1 = 1-r$ and $w_2 > w_1$, where w_2 is the weight for benefit. Then the top m time-horizons from the resulting ranking based on least weighted Euclidean distances would be selected as the suitable time-horizons to trade, where m is some arbitrary integer. It is worth mentioning that when we run this strategy on a few stocks, the actual percentage returns had tendency to be less volatile when more weight was put on uncertainty. Further we need to only trade time-horizons above $y=0$ line, or for short selling, below the line.

9.4 Experimental Details

In the following sections we first detail the purpose of the experiment(s), data used and any additional design details. Section 9.5 will show results on the basis of the following.

9.4.1 Experimental Objectives

The main objective of our experiment is to show that there is some pragmatic benefit in using RQG method. We use a naïve trading strategy to demonstrate the utility of our proposal.

9.4.2 Data

The data used in our experiments were the daily split adjusted, close prices from Wolseley, a FTSE-100 company. Data was separated into training set, validation set and test set, as indicated on Figure 4.

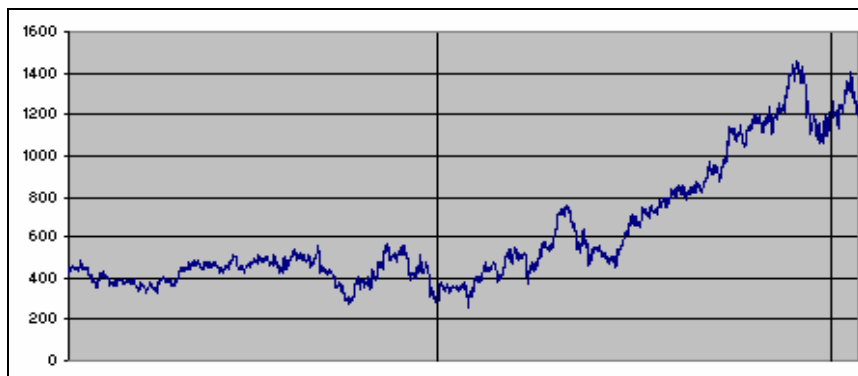


Fig. 4. 10 years of Wolseley time-series data (training/validation/test sets, respectively)

9.4.3 Experimental Set-up and Methodology

We implemented a simple naïve trading strategy based on RQGs for both prediction models (Browns and kNN) described in section 9.2. Figure 5 illustrates the trading strategy; for a particular day j_i an RQG is generated showing predicted output and uncertainty over the next two days. We buy the stock at j_i and either sell on j_{i+1} or j_{i+2} , based on the shortest Euclidean distance as described in section 9.3. We do this for 19 consecutive trading days, at each day we must buy and sell on either of the next two days. In section 9.5 we present returns in percentages for this.

Martin Sykora and Sameer Singh

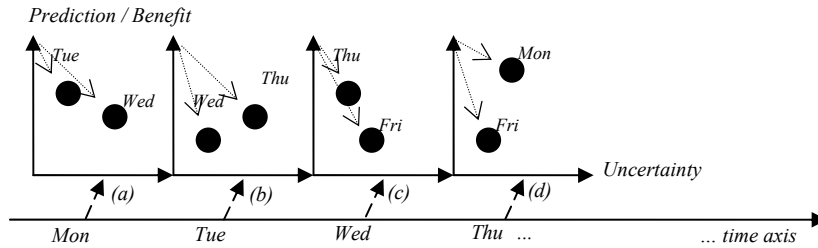


Fig 5. Naive 2 day strategy, predicting on Monday, Tuesday, Wednesday, Thursday...

9.5 Results

Percentage returns for our naive trading strategy are presented for Wolseley, for the period 8th Jan 07 – 2nd Feb 07.

Buy Date	Sell Date	Profit (%)	Buy Date	Sell Date	Profit (%)
05/01/2007	08/01/2007	-0.78	05/01/2007	08/01/2007	-0.78
08/01/2007	10/01/2007	0.24	08/01/2007	10/01/2007	0.24
09/01/2007	10/01/2007	-1.55	09/01/2007	11/01/2007	0.54
10/01/2007	12/01/2007	2.59	10/01/2007	12/01/2007	2.59
11/01/2007	15/01/2007	1.85	11/01/2007	12/01/2007	0.46
12/01/2007	16/01/2007	-0.08	12/01/2007	16/01/2007	-0.08
15/01/2007	17/01/2007	-1.81	15/01/2007	17/01/2007	-1.81
16/01/2007	18/01/2007	1.38	16/01/2007	17/01/2007	-0.38
17/01/2007	19/01/2007	3.77	17/01/2007	19/01/2007	3.77
18/01/2007	22/01/2007	1.96	18/01/2007	19/01/2007	1.96
19/01/2007	22/01/2007	0	19/01/2007	23/01/2007	0.74
22/01/2007	23/01/2007	0.74	22/01/2007	23/01/2007	0.74
23/01/2007	25/01/2007	-2.06	23/01/2007	25/01/2007	-2.06
24/01/2007	26/01/2007	-2.37	24/01/2007	26/01/2007	-2.37
25/01/2007	26/01/2007	-0.97	25/01/2007	26/01/2007	-0.97
26/01/2007	30/01/2007	0.37	26/01/2007	30/01/2007	0.37
29/01/2007	30/01/2007	0.15	29/01/2007	31/01/2007	-0.23
30/01/2007	31/01/2007	-0.38	30/01/2007	31/01/2007	-0.38
31/01/2007	02/02/2007	2.12	31/01/2007	02/02/2007	2.12
Total:		5.18	Total:		4.49

Table 3. Profits (in %), for kNN on the left Browns on the right

Chapter 9 - Developing Trading Strategies based on Risk-analysis of Stocks

As expected kNN, performs slightly better than Browns model. A simple buy-and-hold strategy for the same period would provide 5.15%, our kNN does slightly better. The RQG method has potential, as there seems to be some benefit in using RQG with even a very simple naïve trading strategy.

9.5 Conclusion

This paper proposes an innovative way of looking at risk. Several trading strategies were proposed based on the risk-quantification plots. The forecast system used is central to the accuracy of RQG and the success of any related trading strategies. The better the forecast system the more accurate will our proposed risk quantification be. We believe the method discussed is showing potential for further study within future work on visual risk quantification methods of stock markets and related financial assets.

References

1. Grinold R.C. and Kahn R.N. (2000) Active Portfolio Management, McGraw-Hill, ISBN 0-070-24882-6
2. Pilbeam K. (1998) Finance & Financial Markets, ISBN 0-333-62945-0
3. Hair, Anderson, Tatham and Black (1998) Multivariate Data Analysis, Prentice Hall, ISBN 0-138-94858-5
4. Lindsay D. and Cox S., Effective Probability Forecasting for Time series Data using standard Machine Learning Techniques, ICAPR 05 proceedings, pages 35-44
5. Delurgio S. (1998) Forecasting: Principles and Applications, McGraw-Hill, ISBN 0-256-13433-2
6. Makridakis S., Wheelwright S.C. and Hyndman R.J. (1998) Forecasting: Methods and Applications, John Wiley & Sons, ISBN 0-471-53233-9
7. Trippi R. (1995) Chaos and Nonlinear Dynamics in the Financial Markets, Irwin, ISBN 1-557-38857-1
8. Han J. and Kamber M. (2006) Data Mining: Concepts and Techniques, Morgan Kaufmann, ISBN 1-55860-901-6
9. Burns R. (2005) The Naked Trader: How anyone can make money Trading Shares, Harriman House Publishing, ISBN 1-897-59745-2