Government Office for Science

Foresight

Technology trends in the financial markets: A 2020 vision

The Future of Computer Trading in Financial Markets driver review – DR 3

Contents

Summary 4

1. Overview 5

2. A Brief History of Technology in the Financial Markets. 5

2.1 The First Three Hundred Years: Paper, Horse, Pigeon, Wire, Computer 5

2.2 The Last Decade: Rise of the Robot 8

3. The Current State of Play 14

3.1 Latency issues drive co-location 14

3.2 The need for speed drives a shift to custom hardware 17

3.3 New opportunities drive development of new software techniques 20

3.4 Fresh opportunities, fresh risks 23

3.5 Discussion 24

4. A Decade Hence: the View to 2022 26

4.1 Caveat Emptor 26

4.2 High Performance Computing in the Cloud 27

4.3 Proximity Servers Replaced by Proximity Silicon 28

4.4 Adaptive Algorithms Untouched by Human Hands 29

4.5 The Longer, Wider View 32

4.6 End-game: Sunset on London and a sinking in the wharf 33

4.7 Cyber-Security 33

5. Summary 37

Acknowledgements 37

Appendix: Glossary of Acronyms 38

References 41

Technology trends in the financial markets: A 2020 vision

Prof Dave Cliff, UK LSCITS Initiative, Computer Science, University of Bristol.

Dr Dan Brown, Computer Science, University College London.

Prof Philip Treleaven, Computer Science, University College London.

This review has been commissioned as part of the UK Government's Foresight Project, The Future of Computer Trading in Financial Markets. The views expressed do not represent the policy of any Government or organisation**.**

Summary

The global financial markets have been aggressive early-adopters of new technologies for most of their history. In the past quarter of a century, since the instigation of the “Big Bang” switch to paperless electronic trading, the City of London has led the world in the adoption of new information and communications technology (ICT) for the provision of electronic trading facilities, and the associated distribution of data and news feeds. This hunger for new technologies looks unlikely to be diminished in future.

As well as many opportunities, ICT development has additionally brought risks (some of which are non obvious and even counter-intuitive) for which there is an immediate requirement for careful and thorough evaluation.

New technologies may come in the form of new hardware, new software (including algorithms), or (most likely) combinations of the two. As new technologies become available and more widely adopted, they may significantly alter what market actions and activities are possible, and in the longer term they may significantly alter the socio-economics of the financial markets, and hence also the necessary regulatory and political frameworks that financial institutions operate in.

In this document we establish the historical context for technology adoption in the financial markets, review current technology trends, and then extrapolate them out by five to ten years, in an attempt to identify what the financial-markets technology landscape might reasonably look like in 2020 or 2022. By identifying current products and services that appear to meet the technical definition of *disruptive technologies*, we explore what likely ICT developments over the next ten years will become the most significant to the financial markets, and how those developments might change the industry and affect the employment distribution of human traders. We then briefly speculate on the consequent possible impacts on systemic financial stability.

Our primary contention is, unsurprisingly, that without external regulation and intervention, the shift to trading that is dependent on high-speed high-bandwidth automated adaptive technology (a move that is already well underway) looks set to continue over the next decade. Whilst this in itself may not be undesirable, we should look at influencing the direction of travel. The “deverticalization” of financial trading institutions that was initiated in Europe by the initial MiFID[1] legislation also looks set to continue. It is reasonable to expect that the net result of these trends will be a lowering of barriers to entry, and a significant reduction in the number of employees required by major financial institutions. This “depopulation of the trading floors” could lead to a situation where, by 2020, the long-established primacy of London as a major global trading hub is seriously threatened by clusters of automated trading systems operational elsewhere on the planet, in countries that have hitherto not been major centres for the financial markets.

In addition to the threat of commerce migrating away from London is the threat of increased risk coming our way. Whilst the ecosystem of trading strategies may be relatively varied, the range of risk mitigation strategies is alarmingly more limited. A combination of institutional risk systems existing in lockstep has covertly led to an increased danger of national catastrophe in the event of financial shock.

[1] MiFID stands for Markets in Financial Instruments Directive. A glossary of acronyms used in this report is given in the appendix.

Perhaps the greatest challenge facing the financial services industry, regulators and government is the quantification of 'risk' at all levels but especially in the new forms of derivatives, new algorithms deployed in automated trading, and the systemic phenomena of so-called "flash crashes". New tools for quantifying risk are urgently required; the resources of the academic community should be harnessed, and a new risk management culture should be established in financial institutions.

1. Overview

This document, like all of Foresight's Driver Reviews, is aimed at readers who are not necessarily experts in the field. For non-experts, we offer Section 2 as a rapid tour through the history of technology in the financial markets. Section 3 is where we summarize the current state of play, before moving onto our discussion of possible futures, out to the year 2020, in Section 4. Our discussion of future scenarios is motivated by an analysis of current likely *disruptive technologies*. Here, the word "disruptive" has a particular technical meaning. For readers unfamiliar with the notion of disruptive technologies, we present a brief summary at the end of Section 3.

2. A Brief History of Technology in the Financial Markets.

Let's start by recapping the entire history of the technology of the financial markets, but let's keep it really brief. The story spans all of the 18th, 19th, and 20th centuries, which we deal with over a couple of pages in Section 2.1, and then goes into more detail on 21st century developments in Section 3.2.

2.1 The First Three Hundred Years: Paper, Horse, Pigeon, Wire, Computer

The first joint-stock companies, issuing shares to finance major (and often seafaring) ventures, were formed in the late seventeenth century. The birth of the first shareholder economies in Europe in the early years of the eighteenth century is described by Neal (2000); that tale, and the story of the first major financial crisis, the South Sea Bubble, is entertainingly documented at length by Balen (2002). For over a century, London's trading in commodities and shares was centered in coffee-shops around Lombard Street and Change Alley; legislation to regulate these informal equity markets was introduced after the South Sea Bubble crash and some subsequent insider trading scandals, but the London Stock Exchange was not formally founded until 1801. Throughout this period, for well over a century, there was one primary communications technology that the financial markets were dependent on. That was the horse.

Messages concerning market-moving events such as wars, or the arrival or loss of ships, were relayed from the battlefield or the port back to the location of the markets and exchanges by teams of horse-riding messengers. Financiers owned and operated private networks of riders and horses: the traders with access to the best horses had the best, most current, information and could profit from exploiting it before the news they carried became known to others. Horses were fast, but pigeons were faster: the founders of the Reuters news service used carrier pigeons to relay messages over long distances with greater speed, and at less cost, than horse-borne news. Famously, the London-based trader Nathan Rothschild had carrier pigeons sent to the scene of the Battle of Waterloo; they were released immediately after Napoleon's surprise defeat, and Rothschild learnt of the British victory long before anyone else in London (even the UK Government), which allowed him to buy large amounts British bonds shortly before they soared in value on the news that Britain was now the dominant nation in

Europe.

So, money could be made from low-latency communication, because of the information advantages it offered. Furthermore, the lack of any reliable long-distance communications technology meant that major trading institutions, such as merchant and investment banks, needed to locate their offices physically close to the main markets and exchanges. This need for geographic proximity gave rise to the clusters of financial institutions in major cities such as those around Wall St in New York and London's "Square Mile" centered on Lombard Street. These clusters are still evident today, but advances in financial-market information and communications technology (ICT) mean that their longevity is no longer guaranteed, a point that we return to in Section 4.6.

But London was not the only British city in which shares were traded. In 1964, there were at least 22 legally separate stock exchanges in the UK: major cities such as Belfast, Birmingham, Bristol, Cardiff, Edinburgh, Glasgow, Leeds, Liverpool, Manchester, Newcastle and Swansea all had their own stock exchanges, mostly established in the early 1800's, which prospered with high degrees of autonomy for as long as communication over long distances remained difficult.[2]

Networks of messengers on horseback were rendered obsolete in the second half of the 19th century by the arrival first of the telegraph, and then the telephone.[3] In the UK, improved telecoms led to the gradual decline of the stock exchanges outside of London; in the north of England the various city exchanges merged to form the Northern Stock Exchange, but its operations drew to a close in the first half of the 1970's.[4]

Telephone-based communication was the dominant enabling technology of the financial markets for the first 70 years of the twentieth century. That, and printed paper.

Michael Bloomberg, in Chapter 7 of his autobiography (2001), describes how the information handling system at his then-employers Salomon Brothers in the early 1970's was essentially the same paper-based system that had been operational at the firm's foundation in 1911. Bloomberg pioneered Salomon's introduction of an online computer-based information system, developing a lot of the technology in-house but also buying screen-based information systems such as *Quotron* terminals from external suppliers, linking them to the company's existing IBM mainframes and to the then-new technology of minicomputers sold by competitors to IBM such as Digital Equipment Corporation (DEC). Famously, Bloomberg went on to found his own eponymous information technology (IT) company in 1981 and made a huge fortune selling advanced information terminals to pretty much every major trading institution around the world. Bloomberg was not alone: similar terminals were offered by Dow Jones and by Reuters, among others.

As minicomputers were replaced by personal computers (PCs), and as the computing power and storage capacities of PCs increased while their real costs fell, by the late 1980s it was

[2] E.g., the Irish Statute Book of 196 recognized 22 independent stock exchanges operational in the UK at that time: within decade, onl the London Stock Exchange remained. http://www.irishstatutebook.ie/1964/en/si/0005.html

[3] Stephenson (1996) entertainingly tells the story of the rush to lay transoceanic telecoms cables in the late 1800's, and contrasts it with the rush to wrap the world in optical fibre a century later.

[4] Curiously, there have been recent moves to re-establish regional stock exchanges across the UK, to help provincial companies avoid the costs of raising funds via listing in London. Investbx, the exchange founded in Birmingham in 2007, has not been huge success (see Mundy, 2011), but this has not dampened enthusiasm for a Northern exchange in Leeds (see www.yorkshirepost.co.uk/business/business-news/city_makes_pitch_for_northern_stock_exchange_1_2582429.)

commonplace for traders to be interacting with the markets not only via phone to other traders, but also via a PC terminal for information, for risk management software, and for trade-processing software. By then, the market's reliance on ICT was already critical. Loss of the terminals, the PCs, or the phone system, could totally disable activities on the trading floor of any investment bank.

During the mid 1980's, the PCs that had found their way onto trading floors primarily as information-provisioning systems were put to other uses. The information that the PC was showing on its screen was readable not only by traders but also by any program running on the computer. It was straightforward to write a program that could monitor the price of a financial instrument and flash a warning on the screen to tell the trader to sell that instrument if the price rose above some threshold or "trigger" price; or to sell if the price fell below the trigger. Rather than have the trader's computer terminal flash a warning that then prompted the trader to issue a buy or sell order to the market via that same computer terminal, it was manifestly more efficient to simply have the computer actually issue the electronic order direct to the market. While this was faster and cheaper, it carried some hidden systemic risks: now trading decisions were being taken by machines, armed with little more intelligence than the thermostat that controls a house's heating system; when lots of these thermostatically-stupid automated trading systems, each owned and operated by different institutions or different traders, were allowed to interact in a major market, it turned out that their system-level interaction dynamics could be highly undesirable. After the event, the "Black Monday" stock-market crash of October 1987 was widely attributed, at least in part, to dropping prices hitting the trigger-points of these simple automated trading systems, and thereby causing them to sell. As they sold, so their sales depressed prices further, thereby triggering yet other automated systems to sell, pushing the price even lower, triggering others to sell, and so prices spiralled rapidly downwards into freefall. This, and the broader widespread failure of 1970's and 80's academic research in Artificial Intelligence and "logic programming" to deliver convincingly intelligent solutions or technologies, meant that automated trading in the financial markets entered something of a dark-age period after 1987, a "nuclear winter" that endured for roughly a decade.

Nevertheless, over the ensuing decade, as the real cost of computing continued to fall, management of investment funds became increasingly technical, increasingly dependent on computationally intensive mathematical models to reduce or offset portfolio risk, i.e. to "hedge" the risk in the fund's holdings. So-called *statistical arbitrage* (commonly abbreviated to "stat arb") strategies would identify long-term statistical relationships between different financial instruments, and trade on the assumption that any deviations from those long-term relationships are temporary aberrations, that the relationship would revert to its mean in due course. One of the simplest such "mean-reversion" strategies is *pairs trading*, where the statistical relationship that is used as a trading signal is the degree of correlation between just two securities. Identifying productive pair-wise correlations in the sea of financial-market data is a computationally demanding task, but as the price of computers fell, so it became possible to attempt ever more sophisticated stat arb strategies.

Famously, in September 1998, the Long Term Capital Management (LTCM) hedge fund announced that its primary fund had generated truly devastating losses. LTCM's fund was so big, and had contractual links to so many major Wall Street firms, that the Federal Reserve Bank of New York feared that LTCM's failure could trigger a contagious sell-off on the New York markets, sharper than that seen in October '87. The Federal Reserve organized an emergency bail-out by the major US investment banks: as Lowenstein (2000) artfully recounts, the rescue of LTCM was literally an eleventh-hour deal, with the agreement being stitched together on a Sunday night, hastened by the concern that a firm announcement was needed before the markets opened on the Monday morning. Some commentators at the time criticized the Federal Reserve's role in the bail-out of LTCM, arguing that it could encourage more risk-taking by large financial institutions in future, giving them the impression that the Fed would intervene on their behalf if their risky strategies didn't pay off; that they could be too big to fail. The Black Monday crash of October '87 had dealt a severe blow to confidence in automated trading systems. And a decade later the failure of LTCM dented faith in mathematically sophisticated technical trading strategies. Despite this, in the first decade of the 21st Century, both approaches have undergone a remarkable renaissance that has been accelerating rapidly from 2003 onwards, and the two approaches have combined in the last five years to radically change the nature of the global financial markets. We will deal with that, in detail, in the next section.

2.2 The Last Decade: Rise of the Robot

Teasing out all the influences and contributory factors that led to and enabled the rapid changes of the past five years or more is not something that we can afford to spend much time on here, and anyway we are certainly not professional historians. Despite this, the birth of the World Wide Web, and the subsequent dot.com boom, can serve here as a convenient marker for the start of the renaissance of automated trading. The rise of the Web gave birth to a slew of start-ups offering online markets for a variety of business niches. Major success stories of the internet boom such as Ebay.com, Amazon.com, and Betfair.com (the latter an operator of gambling exchanges, for the buying and selling of bets) are all online markets, i.e. online exchanges. At the same time that these consumer-oriented companies were set up, people with experience in the financial markets started to establish new web-based marketplaces and exchanges for financial instruments. So-called Electronic Crossing Networks (ECNs), linking the computers of various trading houses without routing via the central national stock exchange, had long been in existence: the first, Instinet, was founded in 1969; and the US National Association of Securities Dealers Automated Quotation (NASDAQ) exchange was launched as "the world's first electronic stock market" in 1971. Nevertheless, the advent of the Web significantly lowered the barriers to entry (that is, the cost of the necessary technology fell, the number of people with the necessary skills increased, and so on) and a number of technology-focused ECNs started to operate online exchanges for securities, currencies, and commodities.

At much the same time, the ongoing exponential decline in the real costs of computer-power meant that it was possible to deploy automated trading systems that had considerably more intelligence than a thermostat. In most cases, this apparent intelligence was based not on the logical reasoning beloved of old-school Artificial Intelligence (AI) research, but instead on rigorous mathematical approaches that were firmly grounded in statistical modeling and probability theory. The new wave of automated systems concentrated on *execution* of a trade: that is, the computer did not make the decision to buy or to sell a particular block of shares or commodity, nor to convert a particular amount of one currency into another: those decisions were still taken by humans. But, once the trading decision had been made, the execution of that trade was handed over to an automated execution system (AES). Initially, the motivation for passing trades to an AES was that the human traders were then freed up for dealing with more complicated trades. As AES became more commonplace, and more trusted, various trading institutions started to experiment with more sophisticated approaches to automated execution: different methods, different *algorithms*, could be deployed to fit the constraints of different classes of transaction, under differing market circumstances; and hence the notion of *algorithmic trading* was born.

Perhaps inevitably, the word “algorithmic” was judged to be too polysyllabic, and very many people in the markets now talk instead of “algos”. Traders also refer to their algo systems as “robots”, despite the fact that they are purely virtual entities. We’ll use “robot” and “algo” as interchangeable terms.

One of the primary motivations for developing AES was as a means of reducing *market impact* on large transactions. That phrase needs careful explanation. In brief: if you, as an individual, sell one share in a major stock, then the price you’ll get will most likely be whatever price you see on the market screens at the time you execute your trade: the act of you selling (or buying) a single share has no impact on the price of that stock, because your transaction is just a tiny drop in the vast ocean of liquidity for that equity. Now, instead imagine that you are a major fund manager, and that for some reason you have decided to sell one million shares of a particular equity. If you were to sell all million of them immediately, basic economic theory tells us that the sudden sharp increase in supply (more shares for sale) will depress the price: that is, your sale would have a negative impact on the price of the share that you’re selling. Unfortunately for you, that basic economic theory tells not only us, but also your potential counterparties, that the price will go down as a consequence of this big trade; and, to guard against this anticipated drop in price, they respond to your indication to sell *en bloc* by quoting you a lower price than the market had hitherto been showing. This, then, is *market impact*: merely revealing to the market that you are interested in selling (or buying) in large volume will mean that the price you’re given, i.e. the prices other traders quote per share to buy from (or sell to) you, is significantly lower (or higher) than the current market price you see on the screen. Put most simply, market impact is when the price moves against you before you can do your deal, simply because of the size of your deal.

Many AES for reducing market impact are based on the observation that if you could divide your one big deal into a number of smaller slices, and then trickle-feed those slices into the market over some period of time, maybe the size of the individual slices would not be market-moving (or would not move the market nearly so much) and so you’d get a better price per share, overall.

Say, for the sake of example, that we give an AES the job of executing a sale of 999,999 shares in the time-window between noon and 3pm. An extremely simple “salami slicer” algorithm might break that into three sell orders, each for 333,333 shares, and execute them at 1pm, 2pm, and 3pm. One problem with this approach is that the order has been blindly split into three equal-sized slices, with no question of how large the size of those slices are in comparison to the rest of the market activity for this share at those trading times. Attempting to sell 333k shares may still trigger market impact effects if trading at a particular hour is thin: say that the volume for this share in the rest of the market is only 650k at the time that one of the slices is executed; trying to unload 333k will increase volume by over 50% and a market impact effect is likely. On the other hand, selling 333k shares during a period when the current volume is 7m represents less than 5% of the market -- in periods of such high trading volume the AES could probably get away with selling a much bigger slice without any impact effect.

This trickle-feeding via volume-sensitive slicing is exactly what a *volume participation* algo does, using predictions of future volume from a statistical model of past trading activity in the instrument being traded. So, to continue the example, say that our statistical model for this equity, based on recent trading history data, predicts that the volume of transactions in this equity will be 3m shares from noon to 1pm, 4m from 1pm to 2pm, and 6m from 2pm to 3pm. This 3:4:6 ratio can be normalized to sum to one (i.e. divided through by 3+4+6=13) and rewritten as 23%:31%:46% – these percentages are the *volume-weightings* that the algo uses, and hence it would attempt to execute the trade by selling 230k units in the first hour (23% of the 1m to be executed), 310k in the second, and then 460k in the last. The volume-weighting means that in the first period when trading is thin, the algo trickles in a proportionately smaller slice of the order, less than a quarter of the total to be executed; but in the final period when trading is predicted to be heavy it executes almost half of the total order.

To see how well the algo has done, we can calculate the ratio of the total value of transactions in the time-window (i.e., the share-price of each transaction multiplied by the volume of shares in that transaction) to the total number of shares traded in that time-window, to compute the volume-weighted average price (VWAP) for that share over that time-window. Algorithms that aim to meet VWAP objectives (commonly known simply as “VWAP algos”) will typically use volume-participation methods to aim for (or even guarantee) that a block trade will be sliced and fed into the market in such a way that the average price per share for the overall block trade will match the VWAP in the instrument over a specified period, usually the same period as the window for the trade: see, e.g., Madhavan (2002), Bialkowski *et al.* (2008). In addition to VWAP algos, other algorithms have been developed and deployed that aim for percentage-of-volume objectives; that aim for combinations of volume- and time-weighted criteria; and combinations of many other objectives. In recent years a plethora of books on algorithmic trading have been released: for some examples, see Pole (2007), Narang (2009), and Gregoriou (2010).

At the same time as AES systems were being developed to reduce market impact, other trading teams were perfecting advanced stat arb techniques for identifying trading opportunities based on complex statistical regularities that lay deep in the data: the price and volume data for hundreds or thousands of instruments might have to be considered simultaneously and cross-correlated, in the search for opportunities similar to the pairs trading of the 1980’s, but typically involving correlation functions computed over many more than two instruments. These stat arb approaches were made possible by significant computational infrastructure for computing the statistical analyses that identified trading opportunities, and also by developments in IT-enabled trading infrastructure, so-called *Straight-Through Processing* (STP: where the entire trading process, from initiation to payments and clearing, is one seamless electronic flow of transaction-processing steps with no human-operated intermediate stages), and *Direct Market Access* (DMA: where “buy side” investors and investment funds are given direct access to the electronic order-books of an exchange, rather than having to interact with the market via a “sell side” intermediary such as an investment bank or broker/dealer). In the past decade, DMA and STP have been joined by a third three-letter acronym: SOR, for *Smart Order Routing*, where orders to buy or sell are automatically routed to the exchange or ECN that offers the best price for that order.

Advances such as STP and DMA could be capitalized upon because of the existence of agreed-upon standard electronic communications protocols for the exchange of financial messages. Two protocols, SWIFT (Society for Worldwide Interbank Financial Telecommunication), and FIX (Financial Information eXchange, a standard developed and operated by the non-profit UK company FIX Protocol Ltd) have been operational since the early 1990’s, with FIX having been recognized by many market participants as the *de facto* industry standard over the past decade. In 2006, FIX Protocol Ltd released version 1.0 of a new protocol, FAST (FIX Adapted for STreaming), which involves less verbose messages than the original FIX, and is designed specifically for high-bandwidth, low-latency trading applications; FAST is now at version 1.2, first proposed in 2009. FIX and FAST are open *specifications* of protocols: each protocol can be *implemented* in various manners, and using a variety of programming languages. The software implementation of the protocol is known in the industry as the “engine”: institutions may choose to build their own FIX engine and/or FAST engine, or they may buy a ready-developed engine from an third-party supplier such as RapidAddition[5], the leading developer of FIX and FAST engines. For further discussion of FIX, see the FIX Protocol Ltd web-site.[6]

To understand current algorithmic trading systems it is useful to understand how a trade is executed in an exchange, the different types of trading, and types of participants.

Dealers generally execute their orders through a shared centralized *order book* that lists the buy and sell orders for a specific security ranked by price and order arrival time (generally, on a first-in-first-out basis).

The trading process, illustrated in Figure 1, can be split into three major areas:

- **Pre-trade analysis**: this is the most common use of algorithms within a trading environment. It encompasses any system that utilizes financial data or news to analyze certain properties of an asset.
- **Trading signal generation**: the next step in automating the trading process is to generate an actual trading signal. This level of automation is often employed by *systematic* asset managers and trading institution.
- **Trade order execution**: finally, algorithmic trading can be delegated to execute trades and place orders in one more exchanges.

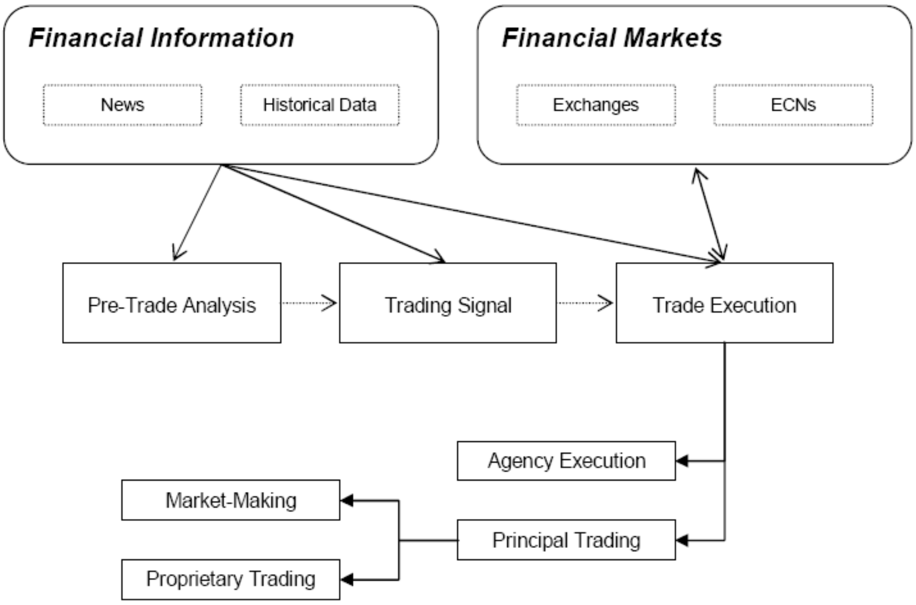

Figure 1: Anatomy of the trading process.

The pre-trade analysis comprises three main components: the alpha model, designed to predict the future behaviour of the financial instruments that the algorithmic system is intended to trade; the risk model, used to evaluate the levels of exposure/risk associated with the financial instruments being traded; and the transaction cost model, which calculates the (potential) costs associated with trading the financial instruments, as illustrated in Figure 2.

[5] http://www.rapidaddition.com.
[6] http://www.fixprotocol.org/what-is-fix.shtml.

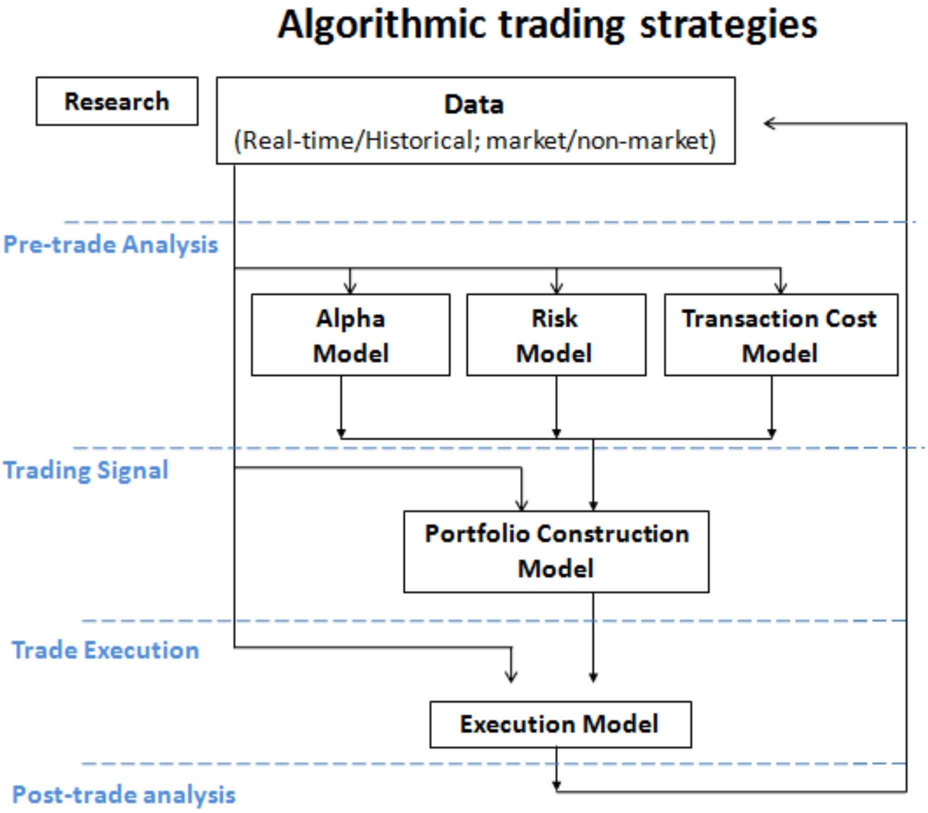

Figure 2: Components of an Algorithmic Trading System

At the Trading Signal stage the primary component of an algorithmic trading system is the Portfolio Construction Model, which takes as inputs the results of the Alpha Model, Risk Model and Transaction Cost Model and optimally selects the ‘best’ portfolio of financial instruments they wish to hold/trade. This involves attempting to maximize potential profits, while limiting risk and the costs associated with the trades.

Finally at the Trade Execution stage, the Execution Model takes the required trades and various data such as the dynamics of the target markets, and executes the trades in as an efficient way as achievable.

With the recent changes in the regulatory frameworks of the financial markets in Europe and in the USA (most notably, the MiFID regulations in the EU), the previously widespread heavily vertically integrated business model of investment banking acting as the “sell side”, selling their privileged access to market liquidity on centralized exchanges to “buy-side” fund managers, has clearly entered a disruptive de-verticalization phase. In consequence, a new business ecosystem of small and medium-sized enterprises (SMEs) has emerged to supply component technologies and new electronic alternatives to existing trading venues and structures. The component technologies are typically “middleware” such as market-data event aggregators, event-stream processors, alpha modeling systems, risk management systems, transaction cost analyzers, automated execution systems, trader’s graphical user interfaces, and so on. The new alternatives to existing trading venues include the “dark pools of liquidity” provided off-exchange by so-called Alternative Trading Systems (ATSs) and Multilateral Trading Facilities (MTFs). The growth of some of these companies has been meteoric. For example, Chi-X is an

MTF established by Instinet in 2007, and now accounts for 26% of FTSE100 equity transactions in London, and 15% of all trading in European shares (Grant & Demos, 2011). Fund-management companies (and also independent individual traders) can now pick-and choose their technology components, and their trading venues, and replicate much – possibly all – of the functionality that they previously paid investment banks to provide. This deverticalization is one major threat to existing long-established business models, an issue explored at length in an influential report co-authored by IBM Business Consulting Services and The Economist Intelligence Unit, titled *The Trader is Dead, Long Live The Trader!* (IBM, 2006).

The convergence of statistically sophisticated and computationally intensive trading strategies, fast automated execution, and direct market access, means that in the last two or three years it has become commonplace for market participants to electronically seek counterparties to a transaction, identify a counterparty, and then execute the transaction, all within a small number of seconds. In consequence, a new style of trading has emerged, known as high-frequency trading (HFT), where automated systems buy and sell on electronic exchange venues, sometimes holding a particular position for only a few seconds or less. That is, an HFT system might "go long" by buying a quantity of shares (or some other financial instrument, such as a commodity or a currency) hold it for perhaps two or three seconds, and then sell it on to a buyer: if the price of the instrument rises in those three seconds, and so long as the transaction costs are small enough, then the HFT system has made a profit on the sale. The profit from holding a long position for three seconds is unlikely to be great, and it may only be a couple of pennies, but if the HFT system is entirely automated, then it is a machine that can create a steady stream of pennies per second, of dollars per hour, twenty four hours per day. A recent study by Kearns *et al.* (2011) indicates that the total amount of money extractable from the markets via HFT may be more modest than some might estimate or guess. Despite this, the low variation in positive returns (the "steady" in "steady stream of pennies") from a well-tuned HFT system is an attractive feature, and one that means HFT is an area of intense interest in the current markets.

HFT hedge-funds have become major players in most of the world's electronically enabled markets. Common current estimates for major equity exchanges around the world are that more than 50% of trading volume is now generated by high-frequency algorithmic trading systems,[7] and recent authoritative projections from the SEC and Boston Consulting Group estimate that this will rise to more than 70% in both the US and Europe by 2015 (SEC & BCG, 2011, p.30).

Finally, no history of the last decade of technology-enabled trading would be complete without mentioning the events of May 6th, 2010, now widely referred to as the "Flash Crash", when the New York equity markets underwent an extraordinary upheaval, with the Dow Jones index losing more than 600 points in a few minutes (representing the disappearance of around one trillion dollars worth of market value) and then regaining much of that loss over the following ten or fifteen minutes. No market crash had ever occurred at such speed, and nor had any rally or recovery. The role of technology in the Flash Crash, and the wider implications of the fact that the global financial markets are now a single interconnected ultra-large-scale complex socio-

[7] A well-informed BBC Radio programme in the *File o Four* series, broadcast in November 2009, covered this in some depth and included lengthy interviews with several senior figures involved in leading-edge high-frequency trading; the broadcast prompted press coverage for the comments made on it by the then HM Treasury minister, Lord Myners. A complete transcript of the 30-minute *File o Four* broadcast is available as pdf from here: http://news.bbc.co.uk/1/shared/bsp/hi/pdfs/20_10_09_fo4_casino.pdf and the BBC News story on Myners' comments is here: http://news.bbc.co.uk/1/hi/business/8338045.stm.

technical system, is discussed at length by Cliff & Northrop (2011). The official report into the causes of the flash crash, jointly published by the US Commodity Futures Trading Commission and the US Securities and Exchange Commission (CFTC & SEC, 2010) argues that it was in fact precipitated by a traditional fundamental trader placing a large $4b sell order. Thus whilst high-frequency algos did not directly initiate the Flash Crash, it seems that they did accelerate the drop, and yet also that they sped the subsequent recovery. An intricately detailed analysis of the precise sequence of trading events on the afternoon of the Flash Crash, which tells a somewhat different story to the official CFTC/SEC report, has been authoritatively published by Nanex Corp[8]: see Meerman *et al.* (2010) and Easley *et al.* (2011) for further details of the extent to which the CFTC/SEC version of events is disputed.

The history story that we've told here is undoubtedly partial in both senses of the word: it's incomplete, and it reflects our personal interests and biases. But this narrative has at least served to introduce the historical context and some core concepts.[9] In many ways the replacement of humans by algos can be seen as the industrialization of trading, analogous to the introduction of robots in car factories to improve the quality of manufacturing.

Now let's turn to how things look today. As it happens, a significant proportion of today's trading technology developments are being driven by exactly the same issue that motivated traders of two hundred years ago to pay for teams of horses and messengers: latency.

3. The Current State of Play

3.1 Latency issues drive co-location

To understand the recent occupation with latency, we need to recap some basic science, and then some basic numbers concerning modern CPU (central processor unit) computer chips.

First, the science.

Computers communicate by sending signals along transmission routes that are typically either electrical pulses along metal wire, or bursts of light along optical fibre, or radio waves transmitted through the ether. Either way, the speed of transmission is limited by the laws of physics, and in particular by the law that no form of electromagnetic radiation can travel faster than a constant *c*, the speed of light. We confidently predict that this law will hold true for the next decade at least.

The speed of light is very fast (approx 300,000,000 metres per second), but in the world of current and future ultra-low-latency trading, it becomes a limiting factor. Trading technology in recent years operated on millisecond timescales, in terms of the time it takes to get a packet of price and volume data from an exchange to the trader's desk, or to match a bid and offer and execute a transaction. Now, in one millisecond (1ms) a signal travelling at the speed of light travels 300km, which is certainly not a huge distance on a planetary scale. The distance from London to New York is roughly 5,500km, and hence the two cities are 18ms away at light-speed.[10]

[8] http://www.nanex.net

[9] For an entertaining first-person account of the developments that we hav   covered here, and for more details on current technology innovations in the financial markets, see Leinweber (2009).

[10] Remember that this is the lowest possible time; in reality there will be delays imposed by the transmitter and receiver hardware at each end of the communications line, any amplification/regeneration steps needed along the way, and by the communication line itself – the constant *c* is the speed of light in a vacuum, but in metal cables the speed is reduced by a "velocity factor" dependent on the

The age of millisecond-scale trading is passing. As we will see later in this review, current trading technology is quoted as operating on microsecond (μs) timescales; in 1μs a pulse of light travels only 300m, so a signal travelling from the London Stock Exchange's HQ in Paternoster Square will take around 18μs to reach the docklands office blocks of Canary Wharf, roughly 5.5km away. And on the story goes: if the technology ever improves to operate on nanosecond timescales, then the fact that in one nanosecond a pulse of light travels no farther than 30cm may become a significant issue.

One consequence of these observations is that in many cases the market data that a trader's computer is currently receiving is already out of date, simply because of the spatial distance that the data has had to cross just to get to that computer. A London trader's screen, showing real-time prices from a New York market, will be at least 18ms out of date in the best case. If the London trader is actually attempting to use the New York data as an input to trading decisions, generating orders that are then to be executed on the New York market, so the latency is doubled: New York data takes 18ms to reach London, and then the trading order from London takes another 18ms to cross the Atlantic back to New York. So, a trader in New York, sat next to the New York market's servers, will have a 36ms latency advantage over the London trader, that's 36ms more time to think and act than the Londoner.

Now 36ms is less than a blink of an eye to a human, but to a run-of-the-mill desktop personal computer (PC) clocked at 3GHz, it is plenty long enough to do serious work.

And so, to the CPU numbers.

A 3GHz clock-speed is nothing special nowadays: PCs with CPU chips running at 3GHz are routinely available in high-street PC stores for less than a thousand pounds. Now, for the sake of this discussion, we'll gloss over some details and just assert that a 3GHz processor executes three billion elementary instructions every second. So, in 36ms it can execute 110 million instructions, 110 million steps of an algorithm. Admittedly, each of these steps is very very simple, but 110 million very very simple steps can nevertheless add up to some extremely sophisticated processing.

If it so happens that 110 million steps is not enough, it may be possible to share the algorithmic processing over some number *n* of multiple independent CPUs, and then combine the results from the *n* CPUs, all within the 36ms time-window. PCs based on single silicon-chip processors with *n* multiple independent processors are also routinely available in high-street PC stores: they are more commonly referred to as "*n*-core" computers; dual-core PCs have been available for several years, and the latest generation of high-end Apple MacBook Pro laptop computers are all quad-core, albeit clocked at no more than 2.3GHz. In 36ms, a quad-core 2.3GHz MacBook can execute over 330 million elementary instructions.

And that is a sufficiently large number to strike terror into the heart of the London trader. However clever she is, however fast her computers are or sophisticated her algorithms are, if she is trying to trade in New York from a desk in London then her data is guaranteed to be at least 18ms out of date when she (or her algo) first sees it, and her order will then take at least

construction of the cable, and in optical fiber the refractive index of the fiber similarly reduces transmission speed Right now, some of the best times for communications latency between New York and Chicago are 13.33ms-14.75ms, offered by the company Spread Networks (www.spreadnetworks.com) who laid their own shortest-path fiber-optic cables between those two cities, digging through mountains to do so (Steiner 2010). The distance between New York and Chicago is 1100km, so roughly one fifth of the distance between New York and London. This implies that actual best-case one-way latency between those two cities would currently be around 70ms.

18ms to reach its destination. The round-trip time of 36ms means that a competitor trader based on Wall St, armed only with a MacBook Pro, can run an algorithm executing a third of a billion simple instructions and still get his order to the exchange before her order has even made landfall in New York. When her order does arrive at the New York exchange, the competitor's order may already have executed, thereby depriving her order of the chance to be executed. The guy in Wall St has eaten her lunch, not because he is cleverer but simply because he is closer.

In this sense then, little has changed since the traders with faster horses in the 18th and early 19th century used their lower latency of information transmission as a competitive advantage. But, for traders dependent on horse-borne messages, there was always the hope of buying or breeding a faster horse, or of developing a new technology that could allow long-distance transmission of messages at speeds much faster than horses could run. The problem presented by the speed of light is that there is currently no plausible route by which the situation can be improved to give superluminal transmission speeds.

The industry's response to this has been to *co-locate* the automated trading machinery with the server computers operated by the major exchanges: now the London trader's computer can talk to a second computer, remotely operated by the London trader, but located in the New York exchange's server building. This remote computer, often referred to as a *proximity server* if it is physically very close to the exchange's servers, and as a *co-lo server* (from co-location) if it is in the same physical data-center, offers the London trader the same latency from the New York exchange's server as the computers of any other traders wanting to connect to that exchange: the exchange that hosts the proximity server will usually do so under a contract that guarantees the same latency for all the co-located proximity servers; metaphorically (and sometimes literally) this is achieved by guaranteeing that each of the proximity servers will be attached to the main exchange server by cables of identical length. Under this kind of arrangement, traders in London and New York each communicate with the New York exchange via the proximity servers hosted at the exchange, and hence each has the same latency of communication to the exchange's main server. Similarly, a London exchange can rent out space for hosting proximity or co-lo servers and thereby allow traders in New York to have low-latency access to the London market. The current arrangement described here is illustrated in Figure 3.

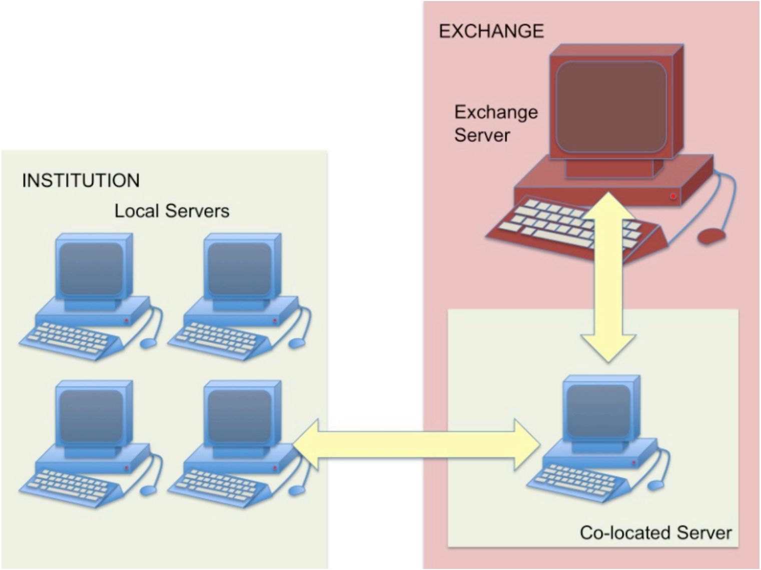

Figure 3: the trading institution's local servers interact with a co-located (co-lo) server, hosted at the exchange; the co-lo server interacts with the central exchange server, minimising latency.

Once everyone has essentially the same communication latency to an exchange server via hosted co-lo servers, the need for speed does not go away: you can be a hundred times cleverer than us, but if we are faster than you, if our order arrives at the market before yours, we still get the chance to steal the deal. With communications latency equalized, the focus shifts to other links in the chain. There are two major ways in which we can try to be faster than you: we can run our algos on hardware that is faster than yours, and we can write algos that are just plain quicker than yours. Faster hardware often involves spending more money, and faster algorithms often involve reducing the number of instructions that have to execute before an order can be issued: in this sense then, Wallis Simpson's observation that you "…can never be too rich or too thin" applies to algos too.

We'll look at faster hardware in Section 3.2, and faster algorithms in Section 3.3.

3.2 The need for speed drives a shift to custom hardware

In the last section we used the capabilities of a current Apple MacBook Pro laptop to illustrate the kind of computing power that is routinely available at reasonable cost. Of course in reality a major investment bank or hedge fund is not going to be running an algo trading operation from a single consumer laptop. Current enterprise information technology (IT) instead often involves running algorithms on "blade servers", a style of computer that is, in essence, a laptop with no screen and no keyboard: just the motherboard in a thin casing, maybe only an inch or so high, which is where the generic "blade" name comes from. The only openings in the casing of a blade server will typically be for a power socket and one or more network sockets: all communication with the blade is via remote network access. In a "rack" unit roughly the size of a wardrobe it is possible to mount fifty or more blades, and their associated power feeds, network cables, and air ducts for cooling. Typically several such racks will be arranged in a line, known as an *aisle*; and in a typical installation there would be one room housing several aisles of racks. The room with the aisles of racks of blades (and associated network switchgear, air cooling/conditioning, power supplies, etc) is the contemporary *data-center*. Superficially, it resembles a commercial mainframe computer installation from the 1960s, but there is an important distinction: those mainframes were single computers; in contrast, a modern data centre of unremarkable size might house several hundred blades, each of which is a motherboard containing at least one, but perhaps two or four, CPU chips; and each of those CPUs would most likely be quad-core at a minimum and may be as high as 16-core. That is, a single blade may be home to as many as 64 independent computer cores, so the typical data-center might be home to hundreds of thousands of independent cores, independent computers, all of them clocked at 2-3GHz, and all of them available to be called upon for automated trading purposes. For several years now, blade-servers have been mass-produced by major manufacturers such as Dell, HP, and IBM.

And there, in that mass-production, lies the rub. To yield the economies of scale that mass-production offers, manufacturers of blade-servers have generally concentrated on offering a relatively small range of relatively general-purpose blade designs: some might have more space on the motherboard occupied by hard disk storage drives (or their solid-state equivalents) for customers who have large data-storage requirements; other blade models might have little or no disk storage but instead lots of RAM (random access memory) chips, for customers who need to manipulate large amounts of data in memory at any one time; yet other blade models may sacrifice RAM and storage to make way for extra CPU chips, for computation-intensive applications.

As banks, fund-management companies, and exchanges became ever more reliant on the capabilities of their data-centers, so the demand for blade-servers that were specialized to their needs increased, and in due course servers “customized” for financial applications started to appear on the market: some produced by the major manufacturers keen to keep their high-value customers in the financial markets; others by small and medium-sized enterprises (SMEs) that recognized the business opportunity in providing technology solutions that the bigger hardware companies either could not or would not provide.

While these custom servers each offer CPU, RAM, storage, and network communications capabilities and configurations that are argued to be better-suited to the demands of the financial institutions than the mass-market servers, they are still servers. That is, they are still fundamentally *general-purpose stored-program* computers: a programmer writes a program, uploads it to the server’s RAM, and the server’s CPU “runs the program” by reading the instructions from RAM and executing them in sequence. While the program is running on the server, that server becomes a specific instance of whatever type of mechanism, whatever type of “machine”, is specified by the program. Say that the program instructs the server to do nothing but calculate the value of π to a trillion decimal places, then the server has become a π-calculating machine, and remains so until the program terminates or the server is reset. After that, loading the server with a different program can turn it into a totally different type of machine: that’s the beauty of computers. The thing is, if you know in advance that really all you want is a π-calculating machine, and you really care about getting the result as fast as possible, then if you can afford it you may be much better off building a *special-purpose* computer, one that exists purely to compute π as fast as possible. Exactly that chain of reasoning has led a number of technology providers to switch to ever-more customized, more trading-specific, hardware. The most dramatic difference between the new custom hardware and even the most trading-specific blade servers is that the new hardware does not involve general-purpose computers, and in some cases may not even involve stored programs in the conventional sense. To keep things brief, we’ll take just two examples.

The first example of custom hardware involves clever repurposing of existing mass-produced hardware that is already heavily customized to a wholly different application area: computer graphics. The sales and marketing material for desktop and laptop PCs often makes mention of the manufacturer of the graphics-board or chip-set inside the computer. Just as the CPU market is dominated by Intel and AMD, so the market for graphics processing unit (GPU) chips is dominated by two companies: Nvidia and AMD. Both companies produce GPUs and associated chips that are, in fact, *n*-core special-purpose computers with their own dedicated high-speed RAM. These are special-purpose massively parallel computer chips that have been designed to execute the computations necessary for 2D and 3D graphics with great efficiency, in parallel, and hence at very high speed.

In the earlier discussion of *n*-core CPUs we spoke of dual-core and quad-core processors; the number of cores in current GPUs are huge in comparison. Because rendering graphics on a computer screen is a task that is readily parallelizable, the *n* in *n*-core GPUs is often much higher: hundreds of cores are simultaneously active in the single GPUs commonly installed in mid-range PCs available on the high street. To distinguish them from the ‘traditional’ 2-core or 4-core CPU chips, processing units with many tens or hundreds of cores are commonly referred to as *many-core* devices. At the top end of the market, single GPUs with more than 500 cores are routinely available, albeit at higher cost, but there is enough demand from players of graphics-intensive computer games that even these are still mass-market, mass-produced chips, and hence relatively cheap. If, for a few hundred pounds you can buy a graphics board that has 500 or more parallel computers on it, so then for a few thousand pounds you can buy a stack of such boards that give you tens of thousands of computers. Each of those computers is very simple, and can only run small programs that were intended to be aimed at painting pixels on a screen, but with some smart programming manoeuvres it is possible to get each of the simple computers in the GPU to do something that is useful in a financial-market context, and their combined effort can yield lightning-fast results that would otherwise require several racks full of blade servers in a data-centre to achieve. One of the world's leading researchers for using GPUs in financial applications is Prof. Mike Giles at Oxford University's Man Centre for Quantitative Finance.[11]

Recognizing the demand for GPU-style cheap but massively parallel simple programmable computers, the GPU manufacturers have very recently started to offer products that support "general purpose computing on GPUs" (GPGPU), i.e. that are less specifically tailored to producing graphics and hence more amenable to deployment in the financial markets and other application areas. New programming frameworks for GPGPU, such as Nvidia's proprietary *CUDA* (e.g. Sanders & Kandrot, 2010), and the emerging de facto industry standard *OpenCL* (e.g. Munshi *et al.*, 2011), are the subject of ongoing development. For a recent readable review of the ongoing shift to GPGPU, see McIntosh-Smith (2011).

The shift from complicated programs running on small numbers of complex CPUs to simple programs running in parallel on very large numbers of simple GPU processors can be considered as a first step away from the historically dominant reliance on general-purpose stored-program computers: the parallel processor cores in a GPU are not general-purpose, but they are still stored-program; they still read instructions from local RAM. The second example of custom hardware that we want to cover is not even stored-program in that sense. Rather than upload a simple program to the local RAM of a simple GPU processor and then run that program, it is possible to have the "program" be the specification of a dedicated processor, a specific electrical circuit of logic gates on a silicon chip. In this scheme, at the moment of execution, the program, the algorithm, does not exist as a sequence of instructions being fetched out of RAM and obeyed by a CPU. Instead, the algorithm is (automatically) converted into a circuit diagram of logic gates beforehand, and those logic gates are then wired together on a silicon chip. When the program is to be run (that is, when the algorithm is to be executed) the circuit on the chip is simply activated and it performs its job without any fetching or executing of instructions: the algorithm has been "cast in silicon". In the early days of silicon chip production, to do this would require huge costs of designing and then tooling up to fabricate a new integrated circuit; costs that could only realistically be borne if the chip then went into mass-production. Fortunately, in the last 15 or 20 years, special mass-produced *reconfigurable* silicon chips have been developed that provide large arrays of logic gates and interconnecting "wires" that can be "programmed" to construct arbitrary circuit arrangements of the available gates, and that can subsequently be "reprogrammed" to configure the gates in different circuit arrangements. These reconfigurable chips are known as *field-programmable gate arrays* (FPGAs).

At the time of writing, only very recently (in the past few months) have FPGA-based trading systems started to be make major inroads into the financial markets. As illustration, we'll highlight two recent stories.

[11] http://people.maths.ox.ac.uk/gilesm/

- FixNetix, a London-based SME established in 2006, announced in November 2010[12] their launch of an integrated FPGA-based automated-execution risk-management system called iX-ecute, which can enforce the pre-trade checks required by trading regulation "in single-digit microseconds", usefully faster than similar non-silicon-based solution from other vendors, which typically operate on timescales measured in tens or hundreds of microseconds.

- In March 2011, Deutsche Bank announced an FPGA-based market access system with 2ms round-trip time: 1.25ms outbound (including risk-checking) and 0.8 ms return of the acknowledgement signal.[13] In an article written by Ivy Schmerken for the *Advanced Trading* news website, Ralf Roth, global head of product development for Deutsche Bank's global equity electronic trading business, made some illuminating comments:

> "We're doing things at wire speed," said Ralf Roth….
>
> Whereas most low latency trading solutions are built in software, Deutsche Bank's solution is based upon a field programmable gate array (FPGA), in which blocks of logic are programmed into a chip.
>
> "This is a bit of a revolution, since it's breaking a barrier from previously doing a couple of hundreds of microseconds and then 80 microseconds which is the normal software-based Ultra products' latency," said Roth. "That is the market standard and now we're getting into the low-single digit microseconds. That has never been done before," he said.
>
> … "The trade comes into the card, the card does the protocol translation and risk checks" explained Roth. "We're bypassing the PC and doing everything in hardware,". (Schmerken, 2011)[14]

The full impact of "bypassing the PC and doing everything in hardware" is likely to be a major driver of change over the next decade; a point that we return to in Section 4.3.

3.3 New opportunities drive development of new software techniques

Current developments in the software of the financial markets can be characterized as concentrating on the very fast, the very big, the very wide, and the very clever. We'll deal with each of these briefly in turn.

Very fast: the desire for low-latency solutions is, of course, not purely a hardware issue; in the software that runs on the hardware, algorithmic sophistication can trump mathematical sophistication. To illustrate this, say that Algorithm X and Algorithm Y use identical mathematics, but (as is often the case) there is more than one way to express the mathematics as a sequence of computer-instructions (i.e., as an algorithmic *implemention* of the mathematics): if X takes 50ms to execute, and Y takes 5ms, then Y wins, despite the fact that the two implementations are equally mathematically sophisticated: By the time X has concluded its computation, Y may already have traded seven or eight times, and the market could then have changed significantly from when X started its computation, so the end-result of X's 50ms of computing is simply no longer relevant by the time it is finished. Similarly, if X involves PhD-level maths and takes 50ms to compute a trading decision that would make a

[12] See http://www.fixnetix.com/articles/display/76/ and http://www.fixnetix.com/services/innovation.html.

[13] http://www.tradersmagazine.com/news/deutsche-bank-high-frequency-trading-microsecond-access-107292-1.html.

[14] See http://www.advancedtrading.com/articles/229300997.

50% profit on execution; but Y involves only high-school maths, and only makes a 5% profit, but only takes 5ms to compute a decision, Y is very likely to outperform X.

In this sense then, a clever algorithm can be beaten by a less-clever one, if the less-clever one is faster. One way in which this desire for speed manifests itself is in the use of rapid accurately approximating ("quick and dirty") stochastic machine-learning techniques as alternatives to statistically rigorous analytic methods; the analytic methods are exact but require the computationally expensive manipulation of vast matrices; the approximations are inexact, but computationally cheap and hence fast. Quick and dirty methods include *artificial neural networks* (see e.g. Oja, 1982; Bishop, 1995; & Haykin, 2008) and *support vector machines* (see e.g. Cristianini & Shawe-Taylor, 2000; Yu, *et al.*, 2009); for an introduction to the broader notion of *probably approximately correct* (PAC) learning, see Valiant (1984); and for an ever wider perspective see Wolpert (1994).

Very big: in searching for new trading strategies, new opportunities to exploit statistical regularities, it is possible in principle to search for correlations (possibly nonlinear ones) between the prices of any *n* financial instruments over some time-window. The sheer number of financial instruments, and the large range of time-windows over which such correlations may need to be searched for, means that truly vast amounts of data have to be computed over – terabytes or petabytes rather than megabytes or gigabytes. New software tools and techniques have been developed for dealing with such "big data" issues. In particular, the *MapReduce* approach at the heart of Google's search engine (Brin & Page, 1998) has been popularized in open-source format as *Hadoop* (e.g. White, 2010), allowing the task of analyzing vast data-sets to be split across many hundreds or thousands of simultaneously active servers. If a financial institution can afford to fill a warehouse-sized building with compute servers, it can now use Hadoop and other open-source tools (see e.g. Janert, 2011) to analyse petabyte-scale financial data sets. If a financial institution, or individual trader, cannot afford a warehouse full of computers then that is not necessarily a problem because such warehouses can now be remotely accessed and cheap rental paid on a by-the-hour basis, an issue we return to in Section 4.2. The mathematics of such analyses has also been developing rapidly in recent years, with a shift away from traditional frequentist approaches which assume that the underlying statistical distributions are analytically convenient ones (such as the normal, log-normal, or Poisson distributions), toward so-called *non-parametric* approaches that make fewer (ideally no) assumptions about the underlying distributions (e.g., Siegel & Castellan 1988; Pett, 1997; Gulati & Padgett, 2003; Ahamada & Flachaire, 2010); and which in the past few years have been firmly integrated with Bayesian, rather than frequentist, statistical approaches (e.g., Hjort, *et al.*, 2010; Ghosh & Ramamoorthi, 2010).

Very wide: here, the "width" is as in band*width.* The number of significant events per unit time (e.g., orders per second arriving at an exchange server) in modern financial-market ICT systems has been growing very rapidly in recent years. Streams of digital events in trading ICT systems are typically now of such high bandwidth, and the individual events are often of such a simple "atomic" nature, that the only practicable approach to dealing with the event-streams is to monitor them in real-time and attempt to identify when particular "complex", compound sequences of events have occurred. That is, there is no hope of running the data out to disk for storage and subsequent analysis "offline", after the stream of data has ceased to flow (an approach that is common in other data-intensive areas such as particle physics, where a single experiment lasting a few minutes may generate petabytes of data for subsequent analysis), because in the financial markets the relevant data streams just never stop flowing for long enough, if at all. In the past half a decade or so, companies such as Progress Apama[15] and Streambase[16] have become market-leaders in low-latency capital-markets applications of what is variously referred to as complex event processing (CEP), event-stream processing (ESP), and business event processing (BEP). For varying perspectives on CEP/ESP/BEP, see Luckham (2002); Taylor *et al.* (2009); Etzion & Niblett (2011); and Cugola & Magara (2011, forthcoming).

Very clever: while we certainly don't intend to imply that the fast, big, and wide software solutions that we have very briefly reviewed in the previous three paragraphs are *not* clever, but there are some interesting recent developments that do not easily fit into any of those three categories. In fact, of course there are very many more such developments than we can reasonably cover in the space available here. So, we'll briefly mention just three exemplars that illustrate where future trading systems currently seem to be headed.

- First, Pipeline Financial[17] have in recent years risen to prominence in the provision of scientifically innovative developments of trading technology. In particular, their development of sophisticated "predictive analytics" (advanced statistical analysis tools), coupled with a truly innovative "algorithm switching engine" which reads the current situation in the market, predicts the near-term-future, and chooses one algorithm (from a large set) that is best-suited to the current and predicted future market conditions, represents a major advance. For further details of the science behind the Pipeline Switching Engine, see Stephens & Waelbroeck (2009).

- Second, it is widely known within the markets that in recent years the major newswire providers have been exploring methods for enabling algorithmic trading systems to make their trading decisions based not only on the numeric data, such as price and volume, present (and readily computer-readable) in market newswire feeds, but also on the linguistic, semantic data that is not naturally expressed in terms of numeric values. Rather than attempt to understand the content and implications of single stories, recent research in this area has been concentrated on the "hive-mind" effects of aggregating over sizeable numbers of stories, preferably from independent sources such as different newswire services. In this kind of approach, there is no intent to "understand" the trading implications of individual newswire stories; rather, the intent is to gauge overall *market sentiment* from a collection of stories, over some period of time, concerning a particular financial instrument: for an up-to-date review of work in this area, see (Mitra, 2011).

- Third, and probably most long-term, comes software research motivated by the common knowledge that correlation is not the same as causation. While great strides have been made in advancing the state of the art in techniques for machine processing of ultra-high-dimensional, multivariate nonlinear (and nonparametric) correlation analyses, driven largely by the needs of the StartArb community, current techniques for computerized establishment of causal links, and for reasoning about causation, are much less well developed. Statistical approaches to establishing causation (for a very specific definition of "causation") in time-series data were developed by the Welsh economist Clive Granger, for which he was awarded the 2003 Nobel Memorial Prize in Economic Sciences. Details of Granger-Causality, and the related issue of cointegration, are described clearly by Alexander (2001). More fancifully, in 2001 Tim Berners-Lee *et al.*

[15] http://web.progress.com/en/apama/
[16] http://www.streambase.com/
[17] http://www.pipelinetrading.com/

wrote a much-discussed article in *Scientific American*, laying out a future vision for “The Semantic Web”, where the computers moving data around the World Wide Web become able to detect and reason about the content, the *semantics*, of the data that they store, receive, or pass from place to place. The vision of developing a “Semantic Web” immediately fired many academics into action (see e.g. Fensel, *et al.,* 2003). If ever they manage to develop such semantic reasoning technology, it could have a transformational effect on the automation of the financial markets. Despite the potential for delivering a lucrative Holy Grail, there are deep philosophical reasons for suspecting that the aims of the Semantic Web research community are simply unattainable (see e.g. Dreyfus, 1981; 1992). A decade ago, it seemed plausible to proponents of the semantic web that their primary aims might be achieved within a decade; one decade later and nowadays it is no surprise to hear a semantic web researcher say that all they need is just ten more years. ‘Twas ever thus. Let’s get back to the reality of the coming decade.

3.4 Fresh opportunities, fresh risks

Failure to quantify risk correctly is now seen to be at the heart of most economic crises (e.g. the subprime counterparty risk crisis of 2007/08) and failed trading firms (e.g. the LTCM collapse of 1998).

In finance, risks are plentiful – complex financial instruments, panic selling, market emotion, and algorithmic trading that feeds on itself. The potential for massive risk is exacerbated by:

- **Highly liquid & global markets** – raise the trading volumes of equities, futures, derivatives, bonds, foreign exchange traded.
- **HFT** – automated trading systems interacting with exchanges and ECNs can issue very many more orders per unit time than were ever seen in the human-led markets.
- **Complex derivatives and structured instruments** – makes the evaluation of risk more complex; for example, new exchange traded funds (ETF) are multi-asset instruments which makes their trading more complex and hence a technical challenge for traders; exotic derivatives often require complex predictions as to future correlations and volatilities, and their hedging rules are highly model dependent.
- **Automated trading** – algorithmic trading is proliferating; in the U.S., high-frequency trading firms represent 2% of the approximately 20,000 firms operating today, but account for over 73% of all equity trading volume.
- **New forms of trading** – for example, so-called “Dark Pools” are a type of trading platform that allow large blocks of shares to be traded without the prices being revealed publicly (to other traders) until after trades are completed.
- **New ways of sharing the spread** – for example, in the past decade so-called “maker-taker pricing” has become commonplace in US equity markets. Under this scheme, an exchange will charge a fee to traders who are “liquidity-takers” (those who place orders at whatever the current market price is) and will pay a commission to “liquidity providers” (those who place “limit orders”, to buy or sell at a specific limit-price or better). It has been argued that maker-taker pricing distorts markets (Angel, Harris, & Spratt, 2010), such distortions offer additional sources of risk.

Traditionally, risk models have focused on the risk associated with a target portfolio of financial instruments and attempted to quantify both the risk associated with individual instruments and with the portfolio as a whole. Only recently have systemic economic risk factors been modelled: this is the modelling of external risk factors such as regime change (reversal of an historic market trend); exogenous shock (major incidents external to the markets such as a war or terrorist incident); and endogenous risk (the market turmoil that occurs when market players believe trouble is ahead and they take actions that bring about realized volatility). An additional and closely related form of identifiable external risk is contagion. Classical contagion involves an economic crisis in one market either actually spreading to other markets, or market participants merely fearing that this will happen and changing their behavior accordingly. Another form of contagion could be said to occur when very many traders (human or algorithmic) all implement the same trading strategy, thereby unintentionally reinforcing a trend; or indeed when they all rely on identical risk models, in which case any omissions in the risk model is amplified to become systemic factors.

The risks associated with high-frequency algorithmic trading are poorly understood, and in turn quantifying the overall risk exposure of a single financial institution has become fiendishly difficult. Failure to model risk correctly is now seen to be at the heart of most economic crises and failed trading firms. Increasingly sophisticated risk models are now being incorporated into algorithmic systems. Portfolio risk can be controlled by setting size limits on the portfolio and on the component financial instruments, and also measuring the volatility associated with individual instruments or groups of instruments. A widely used risk-management technique involves calculating the "value at risk" (VaR) on the basis of past volatility of the instruments being held, but the sub-prime crisis revealed the usefulness of this approach to be highly questionable: an issue discussed at length by Nocera (2009).

It is fair to say that, during the run-up to and unfolding of the subprime crisis, the local-vs-global issues were poorly understood not only by the market participants (the traders, their management, and the shareholders in their companies) but also by the market regulators and political authorities. The so-called "Persaud Paradox", that the observation of safety creates risk (where large numbers of market participants take very similar risk-reducing "safe" positions and thereby, via the near-homogeneity of their positions, greatly increase the overall systemic risk) was spoken about as something of an idle curiosity when the phrase was first coined in a 2005 *Financial Times* article,[18] but within a couple of years events in the real world had conspired to make its downside effects a rather stark reality.

As discussed, one of the greatest challenge facing the financial services industry, regulators and government is the quantification of risk. This urgently requires the resources of the academic community to be harnessed, and a new risk management culture to be established in financial institutions and regulators.

3.5 Discussion

We've looked at how latency is driving co-location, how co-located servers are being challenged by new specialized hardware that allows "bypassing the PC", and how advances in software make that an entirely reasonable thing to do. This current situation, and the likely future, make the development of new, more appropriate risk-assessment tools and techniques an urgent priority.

Ten or more years ago, these technology-driven issues of today may have been difficult to predict. Difficult, but certainly not impossible.

The future of technology is not entirely opaque: it really is plausible that a decade ago someone could have looked at trends in financial-market technology that had been established in the 1980's and continued in the 1990's, and extrapolated them out to the then-distant future

[18] Se e.g. http://www.ft.com/cms/s/1/c84064da-1661-11da-8081-00000e2511c8.html

of 2010 or 2011. A key observation here is this: ten years ago, hardware technologies such as GPUs and FPGAs, and software systems for adaptive automated execution and advanced data-mining were already in existence. None of the things spoken about here have come into existence *de novo* in the past decade. The key technologies were there to be studied and reasoned about a decade ago. Someone looking hard, and *armed with the right intellectual tools*, could probably have made a fair stab of predicting exactly the current state of affairs, a decade before it occurred.

There is a specific intellectual tool, the notion of *Disruptive Technologies*, which we want to use here. In the next section of this review, we make some specific predictions for how we think the technology-enabled global financial markets may look in 2020 or 2022, and we do that by identifying some current technologies that are not yet commonplace in the financial markets, but which look set to change the game within the next ten years: these are the technologies that we believe will be *disruptive* in the coming decade. The specific technical concept of *Disruptive Technologies* was introduced by Bower & Christensen in a 1995 Harvard Business Review article and then expanded upon in Christensen's very successful book *The Innovator's Dilemma* (1997).

In brief, a disruptive technology is one that currently looks weak or incomplete, unable to affect the business of an established technology-producing company or group of companies, and hence is ignored by that company or companies until it suddenly threatens the established business.

To illustrate this, say that established, incumbent companies serve technology of type T1 to some market M1 of customers whose current technology needs, and likely future needs, are well known and will always be well-met by T1. Companies serving M1 do the right thing, concentrating on giving their existing customers what they want based on their successful T1 technology. When a new, currently-weak technology T2 first appears, it cannot serve the M1 customers and hence is seen as of no threat to the status quo. However, if T2 serves some other market M2, of no interest to the companies serving M1, then other companies will most likely invest in developing T2 to serve the growing needs of customers in M2, and hence T2 will mature and improve over time. At some point, T2 may have improved to the point where it is fully capable of serving the needs of the original M1 market, but is smaller or lighter or cheaper or faster, and hence the M1 customers prefer T2 to T1, and the old incumbent companies suddenly find that there is little or no demand for their T1 products, and yet they have no experience or skills in working with T2, and hence possibly go out of business. In this example, T2 is the disruptive technology.

Christensen's original (1997) book and his subsequent publications have provided a wealth of real-world examples in which, with hindsight, disruptive technologies can be seen to have caused major problems for successful incumbent companies. For a critical survey of work on disruptive technologies since 1997, see Daneels (2004); for recent advice on how to handle disruptive technologies, see Markides & Oyon (2010).

Disruptive technologies do not have to be able to improve beyond the capabilities of the currently dominant technology, they need only to be capable of improving to the point where they serve the needs of the mainstream market previously satisfied by that older, incumbent technology. In Section 4, we discuss a number of technologies that we think likely to prove disruptive to the technology-enabled financial markets in the coming decade.

4. A Decade Hence: the View to 2022

4.1 Caveat Emptor

Making predictions that involve setting precise future dates on specific future technology developments is great if you get it right but can be rather embarrassing when you get it wrong. It's a game that we would rather avoid. Nevertheless, it seems reasonable to explore the consequences of taking business and technology trends that have firmly established themselves in the markets in recent years, and extrapolating them out by another decade or so, to explore the potential impact of disruptive technologies. So, while we firmly believe the story that we write here to be plausible, and indeed likely, we are deliberately imprecise about timing and dates. We paint a picture of a future scenario, and a path for getting there, that in our professional opinion is readily achievable within five years or so, in the absence of any major political or economic factors slowing down the development. Of course, in reality, we expect that political and economic factors will indeed play a retardant role. So the picture of the future that we paint here seems to us to be more likely up to a decade away.

Economic factors can manifestly slow down the rate of new technology development and deployment: although many market-trading institutions preside over astronomical amounts of cash-flow, they are typically fiercely focused on internal efficiency and most technology investments are evaluated on the basis of a three-month profit-and-loss accounting mindset. Even in what appear to be hugely profitable companies, words to the effect of "we'd like to do that, but we don't have the money" are surprisingly familiar.

Moreover, when a technology is readily deployable and the funding is there, there can be social, political factors that significantly slow its uptake. These political factors can be both the "big P" politics of national or international government and regulation, or the "small p" politics internal to an organization. One issue that has perhaps slowed the uptake of FPGAs in trading technology is that the number or people who can program them is really very small, miniscule in comparison to the number of people who can write algorithms in established programming languages like C++, C#, Java, or Python. (Programming FPGAs is very different to writing traditional computer programs, and is notoriously difficult). Furthermore, internal organizational politics have already clearly been a significant factor in slowing the uptake of adaptive automated trading technology on trading floors: the manager of any trading floor could, in principle, have replaced many (but not all) of the traders with automated machinery pretty much overnight at any time in the last five years or so, but to do so would have been seen as a declaration of war by those remaining human traders who were not immediately replaceable by machines: they would most likely have sought other jobs, or sought the removal of the manager, or both. For this reason, much of the ingress of new technology onto the trading floors has been incremental, organic growth, sometimes almost by stealth, filling the gaps opened up by "natural wastage" as humans leave the company or move on to other roles. This institutional inertia has also been a factor in enabling new companies to disrupt the businesses of long-established major incumbents: in the past decade it was sometimes easier to quit working on the trading floor of a major investment bank, start up a HFT hedge fund, and compete directly against the investment bank, than it was to convince the investment bank to establish its own HFT unit: the bank's reluctance to establish an internal HFT unit being fuelled by the desire not to "scare the horses", not to alienate the human traders that might feel threatened by the introduction of automated trading technology that so clearly threatens their jobs.

Caveats duly issued, here are some predictions.

4.2 High Performance Computing in the Cloud

Over the past five years there has been an explosion of activity in the computer industry, focused on what is now widely known as "cloud computing". Cloud computing is where the computing power and data-storage facilities of global networks of huge data-centers, each typically containing hundreds of thousands of blade servers in a dedicated warehouse-sized building, is accessed remotely via high-speed high-bandwidth network connections from standard "access devices" such as PCs, laptops, or mobile phones; and the user pays only a small "rental" fee, charged on a per-minute or per-hour basis, for using these remote facilities.

The name "cloud computing" comes from the observation that for very many applications it no longer really matters how powerful your access device is, how fast its processor is or how big its memory is: so long as the access device has enough power and memory to communicate with the remote data-center, the servers in the data-center can do the processing and storage. And (so the story goes) it doesn't really matter where on earth that data-center is located – it may as well be up in the sky, in the clouds.

There are two key aspects that make cloud computing a major shift in computing, to rival the invention of the PC or the rise of the Internet. These two aspects are elasticity and economics.

First, elasticity. The amount of computing power that is "rentable" by the user is smoothly scalable (the industry term for this is that there is "elastic supply"): if you need 10 servers in the morning, 250 servers from noon to 1pm, and then 1 server for the rest of the day, that's fine; in principle the number of servers you can access can be increased or decreased pretty much like turning up or down the volume on a radio set.

Second, the economics. Cloud-computing data-centers are so vast[19] that truly major economies of scale come into force, and a user may only have to pay a few pennies per hour to access each server or each core. Hence, for only a few dollars per hour, a remote user can assemble what is, in effect, a supercomputer that would cost hundreds of thousands of dollars to construct if the user tried to build and operate one herself. From an accounting perspective, cloud computing moves the cost of IT provisioning from capital expenditure to operational expenditure. As Google (one of the major players in the provision of cloud computing services) remark in their marketing: if all you want is milk, why bother to own a cow?

For an introductory overview of cloud computing, see Cliff (2010); for a historical perspective on the development of cloud computing, see Carr (2008); and for discussion of the challenges of building warehouse-sized cloud-computing facilities, see Barosso & Hölzle (2009).

Of course, it's not quite that simple. There are regulatory and legislative issues which mean that the IT managers in financial institutions care deeply about where their company's data is held: for jurisdictional reasons, they may care deeply about the data being held only on servers in the UK/EU, and may very definitely not want their data to be held on computers in the USA. Cloud-computing service providers are well aware of such concerns, and can offer geographic guarantees in their service-level agreements and contracts. Furthermore, as we saw in Section 3.1, the speed of light means that there will be latencies in the system: for very many applications, these may not matter, but for trading activities, the latencies inherent in communicating with remote data-centers can be prohibitive. Latency would certainly be a

[19] For example, Microsoft's latest cloud-computing data-center in Chicago has floor-space of 700,000 square feet, was built to a budget of US$500m, and is estimated to be able to house 224,000 blade-servers: http://www.datacenterknowledge.com/archives/2009/09/30/microsoft-unveils-its-container-powered-cloud/.

problem if an institution tried to run its automated HFT algorithms "in the cloud", but it is important to remember that not all trading is HFT: there are other modes of trading, such as long-only macro trading, that are not so latency-sensitive. Nevertheless, we feel that the most likely impact of cloud computing on activities in the financial markets in the next ten years will not be in the provision of computing facilities that automate execution, but rather in the provision of cheap, elastically scalable, high-performance computing (HPC) which allows massively compute-intensive procedures to be deployed for the automated design and optimization of trading strategies and execution algorithms. Many major investment banks and hedge funds already own and operate their private data-centers, but they do this for business-critical operations and only a fraction of the capacity of these corporate data-centers can be turned to HPC uses. The ability to either extend existing in-house computer power by adding on cloud-based resources (known as "cloudbursting") or to simply outsource all of the HPC provisioning to a cloud provider, opens up new possibilities that are only just being explored.

Automated design and optimization of trading systems seems extremely likely to grow significantly over the next decade, fuelled by the availability of very cheap HPC. The actual execution algorithms will, as today, run on proximity servers co-located with major exchanges or ECNs/ATSs/MTFs, but we predict that those algorithms will be running on custom silicon rather than standard blade servers, and they will be self-learning adaptive systems, automatically designed by computer. We explore custom silicon in Section 4.3, and automatic design of adaptive systems in Section 4.4.

4.3 Proximity Servers Replaced by Proximity Silicon

We saw in Section 3.2 that in recent months some leading-edge technology developers have started to announce single-digit microsecond trading technology based on FPGAs, that bypass the traditional model of a general-purpose PC running a specific program, and instead do everything in customized hardware. FPGAs are a well-established technology (they have been produced commercially since the mid-1980's) and their capacities and capabilities look set to increase over coming years. Nevertheless, programming FPGAs can be a slow and laborious task: the "program" for an FPGA is most often a formal specification of the circuit to be assembled on the FPGA chip, expressed in a complex hardware description language. Hiring a "programmer" for FPGA applications really means hiring a silicon chip hardware designer, and the time taken to program an FPGA can be a lot longer than the time it takes to write an equivalent program in a conventional computer programming language. Once programmed, FPGAs still typically run at slower clock speeds than application-specific integrated circuits (ASICs), i.e. silicon chips that are custom-designed to be specific to an application, but not "field-programmable" after they have left the factory. Recent academic work (Jääskeläinen, *et al.* 2010) has shown how programs written in the *OpenCL* framework for GPGPU (introduced in Section 3.2) can be compiled into FPGA architectures, which is a promising approach to tackling the difficulty of FPGA programming.

However, there is a newer approach to custom silicon production, currently not yet widely adopted, which seems a very strong candidate for addressing many of these problems with FPGAs, and hence seems a good candidate to be a disruptive technology in the coming decade. The new approach is known as *software-defined silicon* (SDS), a phrase coined by a UK company called XMOS, founded by Prof. David May FRS. In the early 1980s, May was co-designer of the *Inmos Transputer*, a radical microprocessor design, the first to be explicitly targeted at creating massively parallel multi-core processor systems (known then as "Transputer Surfaces"): multiple Transputer chips could be laid out in 2D matrix, each chip communicating with its neighbours to the north and south, east and west, on the matrix; a style of computing architecture known as a *systolic array*: see e.g. Moore *et al.* (1987). A new programming language, called Occam, was invented to enable systolic Transputer surfaces to be programmed at a high level.

Thirty years later, and XMOS are producing the next generation of field-programmable (and hence readily customizable) but fast silicon chips, known as XCore chips, that can be assembled into systolic arrays for massively parallel and ultra high-speed applications. The XCore chips are programmable in a high-level software language much like the C or C++ programming languages with extensions (known as XC) for controlling inter-core communication and the eight independent "threads" of programs running simultaneously on each core, so conventional programmers can write algorithms which are then "compiled down" onto the underlying XCore hardware, without the need to learn a specialized language like Occam.

In a promotional video available from the XMOS web-site, May is filmed saying the following:

> "The benefit to the designer is that he doesn't have to design a complex chip. The time and effort involved in designing almost any kind of chip in hardware is, well we measure it months or years, not days. The time we've observed even relatively inexperienced programmers and designers taking to program the XMOS technology, is often measured in days. And furthermore, of course, the ability to change the design quickly, to iterate it, to revise it, is all there because it is just recompiling software; it takes a minute. To simply change a facet of a hardware design, during the design process, still usually takes hours to re-run the tools to produce the revised design". (May, 2011)

Software defined silicon offers the opportunity to smoothly, and very quickly, go from a description of a trading algorithm in a high-level programming language, to having that algorithm running on a many-core massively parallel high-speed composed of customized processing elements, possibly arranged as a systolic array. We expect this approach to be in wide use by 2020 or 2022 as the style of hardware base for proximity servers. Next, we turn to the nature of the trading algorithms that will be running on them.

4.4 Adaptive Algorithms Untouched by Human Hands

The core of current algorithmic trading systems are, from a technical perspective surprisingly simple in comparison to what is already known to be possible in principle. In particular, most current algorithmic trading systems are quite tightly constrained, with very little adaptivity or "learning from experience", other than the maintenance of statistical models, such as those used for predicting future trading volume in a VWAP algo. We expect that in the next decade there will be a concerted move to more sophisticated algorithms, ones that autonomously learn from their experiences (both positive and negative) in the markets. As the algorithms become more sophisticated, so the jobs of maintaining them and of fine-tuning them to current market conditions, will become more involved. This would be a problem if skilled humans are required to perform the fine-tuning, but the likelihood is that in future the design of new algorithms, and the tuning and optimization of existing ones, will also be an automated process, performed by computers. Rather than hiring programmers to write new algorithms, trading institutions will instead in future hire programmers to write the computer systems that design the new algorithms and then fine-tune their subsequent operation. Once again, we mention this technology not because it is currently able to meet the demands of major customers in the financial markets, but because it seems likely to play a disruptive role in the coming decade.

The heritage of this approach can be seen to stretch ten or fifteen years or more into the past.

In a landmark study published in 2001, a team of researchers at IBM (Das, *et al.*, 2001) presented results from laboratory tests of markets populated by mixtures of human traders and self-learning robot traders (i.e., autonomous adaptive automated execution algorithms), which showed that the human traders were consistently out-performed by two algorithmic systems, and that result was consistent across a variety of experiment designs. IBM's experiment designs were directly inspired by the Nobel-Prize-winning work of Vernon Smith, who established the field now known as *experimental economics* (see, e.g., Smith, 1991; Kagel & Roth, 1997; Smith, 2000; Miller, 2001); IBM used the approach established by Smith to study the interaction of humans and robots in a form of electronic marketplace known in the economics literature as the *continuous double auction* (CDA), which is the auction mechanism at the heart of most of the world's financial markets. The two algorithmic systems that beat humans in the CDA were IBM's "MGD" algorithm (a Modified version of an algorithm first described by Gjerstad & Dickhaut in 1998) and an algorithm called "Zero Intelligence Plus" (ZIP) that had been developed at Hewlett-Packard Labs (Cliff 1997). Both MGD and ZIP "learn from experience", using machine-learning techniques to adjust their behavior in the market to reflect what offers and bids they have seen made by other traders, and which were accepted and which rejected. For further discussion of experimental-economics-style studies of human traders interacting with robot traders, see: Grossklags & Schmidt (2006); Grossklags (2007); De Luca & Cliff (2011a, 2011b); and De Luca *et al.* (2011).

The moment-to-moment trading behavior of algorithms such as MGD and ZIP, and also their longer-term learning behavior or adaptivity, are determined by a small number of "control parameters", which we can think of metaphorically as knobs or sliders on a control panel, each of which runs through a range from 0.0 to 1.0; the operator of the algorithm needs to set each of these parameters, turn each of the knobs, to the right value to get the best behaviour out of the algorithm. But what number constitutes "the right value" may be dependent on the current market conditions. Say there are three control knobs, it may be that right now they should be set to (0.2, 0.8, 0.7), but in different market conditions perhaps the algorithm would perform better if the knobs were set to (0.3, 1.0, 0.2). Optimizing, or at least fine-tuning, the behaviour of an algo like MGD or ZIP requires someone, or something, to "twiddle the knobs" until they find the right value for the particular market conditions that the algo is operating in.

Now, in principle, a human could be paid to twiddle the knobs, to search of good or optimal combinations of values of the control parameters, but humans are famously slow, expensive, and error-prone. As it happens, there are a variety of automated optimization techniques that allow a computer to "twiddle the knobs", i.e. to search the space of possible combinations of parameter values, to find good settings. One very popular automated search and optimization technique, the primary appeal of which lies in its simplicity and its surprising effectiveness, is an inherently parallel method called a *Genetic Algorithm* (GA), a form of "evolutionary optimization", directly inspired by Darwinian natural selection, random variation, and heritability in populations of organisms, (see, e.g. Goldberg, 1987).

In the past decade a number of authors have explored the use of GAs to optimize trading strategies such as ZIP, with significant success. Encouraged by these successes, researchers observed that the number of control parameters could be greatly increased: while a human may find an algorithm with hundreds of control parameters an impossibly daunting task, for a computer optimization process such as a GA there can be little difference between twiddling ten control knobs and twiddling hundreds or thousands. So, more trading algorithms with many tens or hundreds of control parameters are now routinely optimized by automated systems such as GAs or, increasingly commonly, more mathematically sophisticated and computationally efficient optimization methods such as “Estimation of Distribution Algorithms” (EDAs: see e.g. Larrañaga & Lozano, 2001; Lozano, *et al.,* 2010). Researchers have used GA-style or EDA-style optimization not only to fine-tune trading algorithms, but also to fine-tune the auction mechanism, exploring ways whether the CDA mechanism at the core of the world’s financial markets might be improved upon for situations where most or all of the traders in the market are algo systems instead of humans (see, e.g. Cliff, 1998; 2003; 2009; Phelps *et al.,* 2002; Byde, 2003; Phelps *et al.*, 2009). GAs have even been shown to be able to successfully find new designs for FPGA circuits, albeit not in a trading context (Thompson, 1998); an approach known as “evolvable hardware”.

The use of machine optimization methods, such as GAs or EDAs, to design and optimize autonomous adaptive trading algorithms, looks set to increase over the next decade and to act as a disruptive technology. This is a development that is enabled and accelerated by the step-change drop in cost of high performance computing (HPC) offered by cloud computing service providers, as described in Section 4.2. The algorithms being optimized will, we expect, be uploaded to proximity servers based on software-defined-silicon technology for ultra-high-speed execution, as described in Section 4.3. And that is our vision of the technology-enabled trading systems of 2020 or so, a situation we illustrate in Figure 4 from the perspective of a trading institution, and in Figure 5 from the system-level view of the multiple institutions interacting with the exchanges. Of course, the likelihood is that in practice each institution will interact with multiple exchanges.

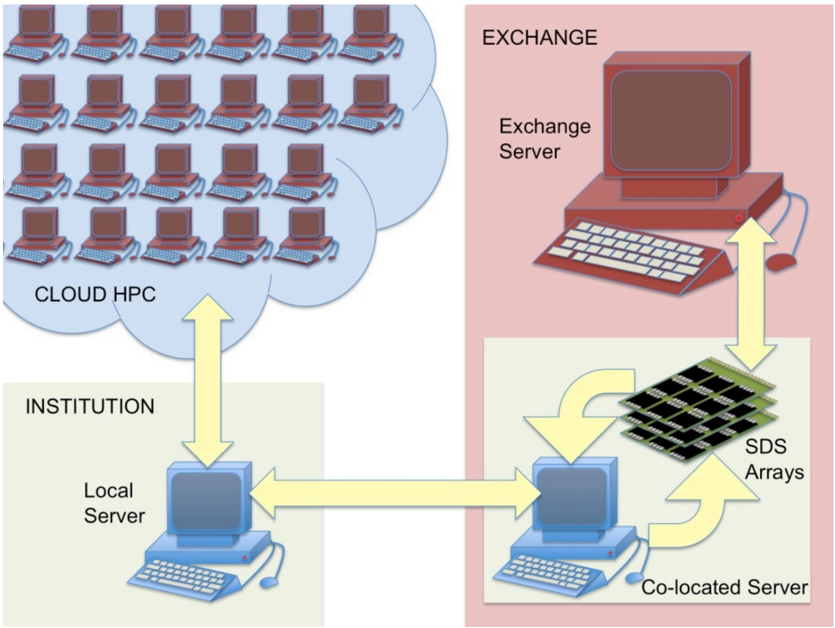

Figure 4: A vision for 2020; the institution’s view. The trading institution’s servers routinely “burst” into the cloud when high-performance computing (HPC) needs exceed the institution’s local capacity. The HPC is used for compute-intensive data-mining and automated design and optimization of trading algorithms. The algorithms are transmitted to the institution’s co-located (co-lo) server, hosted at the Exchange. The co-lo server acts as an interface to the institution’s hosted cluster of software-defined-

silicon (SDS) arrays. The SDS cluster, operating at sub-microsecond speeds, interacts with the main exchange server.

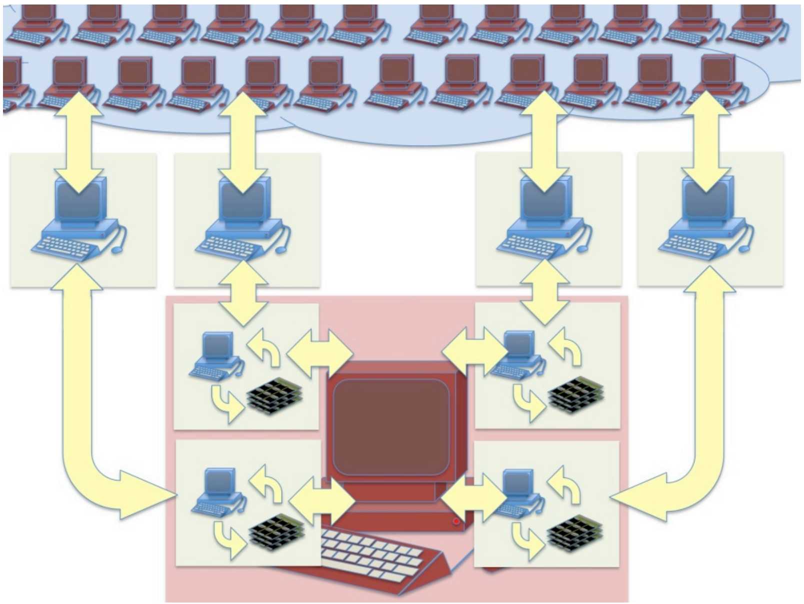

Figure 5: A vision for 2020; the systemic view: multiple institutions each have hosted co-lo servers and SDS arrays at the exchange; the SDS arrays run the processing algorithms, the proximity servers act as local front ends to the SDS arrays; each of the institutions also use cloud-computing services for HPC. Color-coding and iconography as for Figure 4.

4.5 The Longer, Wider View

The technologies we have discussed here are not the only research developments that are likely to alter trading technology in coming years. Academic and industrial research is making significant advances in a number of relevant areas. Some are significant improvements on existing technologies, such as higher-speed telecoms hardware (see, e.g.: Richardson, 2010); others offer the prospect of being truly revolutionary, such as quantum photonic computing (see, e.g. Ladd, *et al.*, 2010), or harnessing living matter for computing in biological substrates such as natural or genetically-engineered micro-organisms (See e.g. Weiss, *et al.,* 2003; Zauner 2005a, 2005b, 2005c). Research fields such as these offer the prospect of producing disruptive technologies, without doubt, but the specification for this review document required us to limit our view to the next decade. Our opinion is that the disruptive effects of these technologies are most likely to be more than a decade away, and so we have not discussed them in any detail here.

Similarly, there may be coming major societal changes such as significant alterations to patterns of energy generation and usage (e.g., MacKay, 2008) or the re-thinking of global monetary systems (e.g., Lietaer, 2002; Hallsmith & Lietaer, 2011) that alter the landscape significantly, but again we do not expect these factors to have a major disruptive effect on the technology of the financial markets within the next decade.

4.6 End-game: Sunset on London and a sinking in the wharf

If our predictions of future developments do pan out, then roughly a decade from now the industry will be at the point where the major trading venues are each serving data to and taking orders from algorithmic trading systems running on nearby co-lo or proximity servers, much as like today. Unlike today, the co-lo servers will not be CPU-based computers running programs, but rather will be based on massively multicore chips such as GPUs, custom silicon such as FPGA, and/or software-defined silicon such as XCore. The trading algorithms that are running on, or embodied in, these nonconventional computing architectures, will have been uploaded to each proximity server from the algorithm research & development (R&D) unit of the trading institution that is the owner of that server (or rentee of the hosting service). The R&D unit will have access to cloud-style elastically scalable large-scale high-performance-computing (HPC) facilities for automated design, evaluation, and refinement of trading algorithms, and for optimizing the trading algorithms to the custom silicon that is available on the proximity server. The R&D unit's HPC provision may be in-house, from the institution's own "private cloud" data-centre, or it may be entirely outsourced to a cloud provider, or perhaps some mix of the two via "cloudbursting". This seems entirely plausible to us.

New approaches to finance are required that more realistically incorporate the institutional environment, regulations and trading behaviour to make economic models better at explaining systematic (non-idiosyncratic) as well as systematic investor/trader decisions, taking into consideration their emotions, the constraints they operate under and their cognitive mechanisms and how these influence decision making. New systems are being developed to analyze market behaviour and the attitudes of financial professionals. As this new approach to finance develops, it is intensifying its use of tools and techniques from quantitative finance, so that mathematical and statistical methodologies are being employed to understand the constraints, incentives and biases of decision makers (fund managers, traders, etc) and their impact on market valuations.

From a UK national strategic perspective, this vision of the future holds some concerns. Consider this question: if this really is how the technology-enabled financial markets will look in 10 years or so, why should any of it be happening in London? For sure, London is by many metrics the leading financial trading hub in Europe, and on the national scale London has been the only city to house a stock exchange since the closure of the last of the UK's regional exchanges in the early 1970's. Perhaps we can better explore the issue by recasting the question: why should any of this future vision *not* be happening in a *different* country?

The current situation in Europe, and indeed much of the rest of the world, is that individual countries still have their own national stock exchanges, the existence of which has often been written into that country's market-trading legislation. But the past five or so years has seen a wave of mergers and acquisitions (M&A) activity that has resulted in the current situation, where very many of these superficially independent national exchanges are in fact owned and operated by one of a small number of multinational conglomerate companies such as NYSE Euronext and Nasdaq OMX. At the time of writing there seems to be an additional wave of M&A as NYSE Euronext and Deutsche Börse have announced intentions to merge, as have the London Stock Exchange Group and TMX, the operators of the Canadian Stock Exchange.

The situation seems interestingly similar to that in mid-20th-century Britain, where long-established and successful exchanges in many of the major cities outside London were subsequently rendered redundant by the establishment of strong telecoms links to London, and the regional exchanges were absorbed by the London Stock Exchange company. It seems plausible that within a few years, the number of independently operational exchanges in Europe could fall significantly. Some current exchanges, that historically have been wholly autonomous, could become mere “mirror sites” showing the trading data from a larger “hub” exchange elsewhere in the EU. It is perfectly possible that one of the larger hub exchanges will be based in London, but it is not impossible that London’s exchanges shrink to become the mirror sites. Such an adjustment of the balance of power among European exchanges is certainly not impossible. And there is yet another possibility, that a totally new player enters the market in a disruptive fashion.

One potential driver for relocating major trading hubs is the consideration of where on the planet it makes most sense to site an exchange, taking into account speed-of-light propagation delays, an issue explored in some detail by Wissner-Gross & Freer (2010).

For the sake of argument, let’s hypothesize the existence of a country somewhere else in the world, referred to here as *Country C*. Let’s say that Country C has built up significant capital reserves in recent years, and is prepared to spend those reserves to become a major player in the global financial markets: a new London or New York or Tokyo. The government of Country C invests in the rapid construction of a national network of cloud-style data-centres, ultra-high-speed fibre communications links between those data-centres and the city designated as the new financial hub, and leading-edge wireless networking across that city. Country C also becomes a major customer for software-defined silicon products, such as XMOS’s XCore. Let’s say that the costs of real-estate, and skilled human labour, are greatly less in Country C than those in the EU; and that Country C has invested in graduate and postgraduate education (either at home or abroad) to ensure that an appropriately educated workforce is readily available, and that these graduates have good skills in English, which a decade from now seems likely to still be the default language of international commerce. Let’s also say that Country C has a more lenient tax structure than is common in European countries, and that it has sufficiently well-developed regulatory and legal systems that governance and compliance are no more of a worry than in the EU. What then? What would stop Country C from destroying London’s concentration of financial-markets businesses?

That last question is one that we are unsure of the answer to, and that we intentionally leave open here, for discussion. We’ll close this section with one more question, again rhetorical: just how many real-world countries are there that could, like our hypothetical Country C, rapidly rise to become a new major hub in the global financial markets within the next decade? We think there are at least two, and possibly more.

One of them begins with C.

4.7 Cyber-Security

Finally, we close this review of future issues with a brief discussion of issues in cyber-security. Electronic attacks on the computer systems and communications networks of the global financial markets are attractive to two communities: profit-seeking criminals seeking to steal money or assets from the system, to enrich themselves; and damage-seeking “enemy agents” who aim to disrupt or destroy the system for reasons other than personal enrichment. The "enemy agents" might be individuals acting alone, terrorist organizations, or nation-states in times of warfare. These two groups, the criminal and the enemy, share some aims and methods but differ in others. Both groups seek to gain unauthorized access to IT systems, to "hack" the system, without triggering any alarms. Once "inside" the system, a criminal hacker would ideally like to operate undetected, stealing as much as possible for as long as possible, and ideally then also to exit the system (to "leave the scene of the crime") without detection. In contrast, an enemy agent may be content to operate undetected only for as long as it took to initiate the desired damage or destruction of the system: after that, there may be much less desire for a clean exit.

It is notable that whilst large-scale nation states may have a significant capacity to attack, they are so much integrated into the globally-connected financial network that any damage inflicted would be likely to reverberate back on them anyway.

While institutions are often understandably secretive about how many cyber-attacks they encounter and the nature of any resultant security breaches, there is no shortage of news articles that indicate the serious nature of the problem. In 2006, the Russian Stock Exchange was closed by a computer virus infection (Sophos, 2006). More recently, in February 2011, the Nasdaq exchange was attacked by hackers injecting malicious programs ("malware") into the system (Stafford *et al.*, 2011; Demos, 2011a; Leyden, 2011a), and closer to later home in the same month the web-site of the London Stock Exchange (but *not* its main exchange servers) was hacked to serve malware hidden in adverts on the site (Leyden, 2011b).

Cyber-security, in finance and any other context, involves an ongoing arms-race between attackers and defenders, predators and prey. Technically, the attackers and defenders are locked in a *co-adaptive dynamic*, in much the same way as co-evolving species are in biological systems. As Van Valen (1973) pointed out, co-evolving or co-adapting agents can show 'Red Queen" dynamics where continuous adaptation is required purely for one population to maintain the same level of fitness or performance relative to the other populations that it is co-evolving with/against – a dynamic named after the Red Queen in Lewis Carroll's *Alice Through the Looking Glass*, a character who is forever running forward, merely to stay in the same place, because the landscape on which she runs is itself rapidly moving backwards under her feet.

It is beyond the scope of this document to provide a review of issues in financial-market cyber-security and counter-terrorism. In November 2010 the UK think-tank Chatham House published a major review, albeit not specific to the financial markets (Cornish *et al.*, 2010), to which we refer the interested reader. While it is of course essential to protect the ICT systems of the global financial markets from exogenous malicious attacks, we are concerned that such attacks seem to have been almost the sole focus of attention by the national security services, the regulatory authorities, and the institutions themselves. The ICT systems that drive the markets can plausibly be disrupted or destroyed by malign intent, and such malicious exogenous attacks should surely be guarded against, but there is also the very real prospect of large-scale systemic failures being caused by *benign endogenous causes*. The everyday participants in the marketplace are expert highly trained fighters involved in a zero sum war, and in many ways, it is the incentive structures and expertise of the endogenous community that is more likely to produce widespread damage and disruption than malicious activities of an exogenous terrorist band seeking to attack the system.

The sort of systemic failures that terrorists might dream of instigating can in fact occur by straightforward combinations of routine component failures and/or unusual operator actions triggering domino-effect chains of failure that ripple out over large chunks of the financial network, causing widespread disruption or highly dysfunctional market dynamics (such as the May 2010 Flash Crash) purely because of the ultra-large-scale, complex, socio-technical nature of the current state of the global financial markets, an issue explored at length by Cliff & Northrop (2011).

5. Summary

In this review we've recapped the historical context for technology adoption in the financial markets, reviewed current technology trends, and then attempted to identify coming disruptive technologies on the assumption that current trends will continue over the coming decade.[20,21]

Without external intervention, the shift to trading that is dependent on high-speed high-bandwidth automated adaptive technology seems set to continue over the next decade. The "deverticalization" of financial trading institutions that was initiated in Europe by the initial MiFID legislation also looks set to continue. It is reasonable to expect that the net result of these trends will be a lowering of barriers to entry, and a significant reduction in the number of employees required by major financial institutions. This "depopulation of the trading floors" could lead to a situation where, by 2020 or 2022, the long-established primacy of London as a major global trading hub is seriously threatened by clusters of automated trading systems operational elsewhere on the planet, in countries that have hitherto not been major centers for the financial markets. In that very macro sense, the stability of the UK financial sector, and London's prominence in the global financial markets, may be about to come under threat. The threat is not inescapable, but continued investment in leading-edge technology research, development, and deployment, and in the "skills base" of appropriately qualified and experienced workers, will be necessary to maintain the position that London has built up, and prospered from, over the past three hundred years. It will also be necessary to engage in programs of research that further develop our understanding of, and ability to quantify, endogenous systemic risk.

Trading systems can today exist anywhere. A benign regulatory framework is required that both entices trading into the UK and provides a beneficial risk management environment. This is a global requirement that the UK should be leading the world on.

There are several industries that the UK used to be globally dominant in, for which a combination of increased automation and globalization have combined to push the British sectors of those industries into decline or collapse. Britain can ill afford that to happen to its position in the global financial markets.

Acknowledgements

Thanks to Philip Bond, Marcus Hooper, Kevin Houstoun, David May, and Simon McIntosh-Smith for valuable discussions and for their comments on earlier drafts of this paper. We also thank the two anonymous peer-reviewers appointed by Foresight, and the comments from the UK Financial Services Authority and Her Majesty's Treasury, all of which were very useful.

[20] Late in the preparation of this review document, we became aware of two other reviews that are each highly relevant, and complementary, to ours: see Angel, Harris, & Spratt (2010) and Gomber *et al.*, (2011).

[21] (Note added in proof:) For a very recent discussion of high-frequency trading, including interviews with leading practitioners, see E. Perez (2011), *The Speed Traders.* McGraw-Hill Publishers.

Appendix: Glossary of Acronyms

This report has used a fair number of acronyms. Here we list them all for quick reference. Each of them was defined or explained the first time they occurred in the text of the report.

AES: Automated Execution System.

AI: Artificial Intelligence.

ASIC: Application-Specific Integrated Circuit.

ATS: Alternative Trading System.

BEP: Business Event Processing.

CDA: Continuous Double Auction.

CEP: Complex Event Processing.

CFTC: Commodity Futures Trading Commission.

CPU: Central Processing Unit.

DMA: Direct Market Access.

ECN: Electronic Crossing Network.

EDA: Estimation of Distribution Algorithm.

ESP: Event Stream Processing.

ETF: Exchange Traded Funds

FAST: FIX Adapted for STreaming.

FIX: Financial Information eXchange.

FPGA: Field-Programmable Gate Array.

GA: Genetic Algorithm.

GPGPU: General-Purpose computing on Graphical Processing Units.

GPU: Graphical Processing Unit.

HFT: High-Frequency Trading.

HPC: High Performance Computing.

IC: Integrated Circuit

ICT: Information and Communications Technology

IT: Information Technology.

LTCM: Long Term Capital Management.

M&A: Mergers and Acquistions.

MGD: Modified Gjerstad-Dickhaut.

MiFID: Markets in Financial Instruments Directive.

MTF: Multilateral Trading Facility.

NASDAQ: National Association of Securities Dealers Automated Quotation.

PAC: Probably Approximately Correct.

PC: Personal Computer.

RAM: Random Access Memory.

R&D: Research and Development.

SDS: Software-Defined Silicon.

SEC: Securities and Exchange Commission.

SME: Small/Medium-sized Enterprise

STP: Straight-Through Processing.

SWIFT: Society for Worldwide Interbank Financial Telecommunication.

VWAP: Volume-Weighted Average Price.

ZIP: Zero Intelligence Plus.

Author Biographies

Dave Cliff

Dave Cliff is a Professor of Computer Science at the University of Bristol. He has more than 20 years of experience as a researcher in computer science and complex adaptive systems. He's previously worked in academic faculty posts at the University of Sussex, at the MIT Artificial Intelligence Lab, and at the University of Southampton. He also spent seven years working in industry: initially as a senior research scientist at Hewlett-Packard Research Labs where he founded and led HP's Complex Adaptive Systems Research Group; then as a Director in Deutsche Bank's London Foreign-Exchange Complex Risk Group. His research for HP included early work, in the mid-to-late 1990s, on novel decentralized management systems for utility-scale "cloud" computing systems; as part of that work he invented the Zero-Intelligence Plus (ZIP) adaptive automated trading strategy. In 2001 a team of researchers at IBM showed that ZIP algorithmic traders consistently outperform human traders; and since then most of his research has been directed at developing new technology for the financial markets. In October 2005, Cliff was appointed Director of the £10m EPSRC-funded five-year UK national research and training initiative in the science and engineering of Large-Scale Complex IT Systems (the LSCITS Initiative: www.lscits.org). He is author or co-author on around 100 academic publications, and inventor or co-inventor on 15 patents; he has undertaken advisory and consultancy work for a number of major companies and for various departments of the UK Government; and he has given around 200 keynote lectures and invited seminars.

Dan Brown

Daniel Brown is Director of Industrial Relations for the Doctoral Training Centre for Financial Computing at UCL. He heads up UCL's Banking Science initiative which has supported the industrialisation of faculty, doctorate, MSc and undergraduate intellectual property in the development of its Trading Platform. He has led on the development of UCLs ATRADE Algorithmic Trading platform used to research and develop algorithms for trading.

Philip Treleaven

Philip Treleaven is Professor of Computing at University College London (UCL) and Director of the UK Centre for Financial Computing: a collaboration of UCL, London School of Economics, London Business School and the leading Investment Banks and Funds (www.financialcomputing.org). The Centre has 35 PhD students working on computational finance, many specializing on algorithmic trading. His research group pioneered the use of quantitative analytics for financial fraud detection (building the first insider dealing detection system for the London Stock Exchange), and for the past six years we have worked closely with a number of Investment Banks helping them to develop their Algorithmic Trading systems. In addition, the Centre has developed a comprehensive and unique Algorithmic Trading platform, used to research and develop algorithms for trading. This platform has been recently used to run an algorithmic trading competition (sponsored by Microsoft) attracting 60 entrants in 36 teams. It is currently being modified so that academics and industry professionals can conduct simulated and real evaluations of algorithmic trading and behavior finance risk.

References

I. Ahamada & E. Flachaire (2010). *Non-Parametric Econometrics*. Oxford University Press.

C. Alexander (2001). *Market Models: A Guide to Financial Data Analysis*. Wiley.

J. Angel, L. Harris, & C. Spratt (2010). *Trading in the 21st Century.* Unpublished manuscript. http://www.sec.gov/comments/s7-02-10/s70210-54.pdf.

M. Balen (2002). *A Very English Deceit. The Secret History of the South Sea Bubble and the First Great Financial Scandal.* Fourth Estate.

L Barroso & U Hölzle (2009). *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*. Morgan Claypool. E-book downloadable from: http://www.morganclaypool.com/doi/pdf/10.2200/S00193ED1V01Y200905CAC006.

T. Berners-Lee, J. Hendler, & O. Lassila, (2001). The Semantic Web. *Scientific American*, May 17th, 2010.

J. Bialkowski, J. Darolles, & G. Le Fol (2008). Improving VWAP Strategies: A dynamical volume approach. *Journal of Banking and Finance*, **32**(9):1709-1722.

C. Bishop (1995). *Neural Networks for Pattern Recognition*. Clarendon Press.

M. Bloomberg (2001). *Bloomberg by Bloomberg.* Wiley.

J. Bower & C. Christensen (1995). Disruptive Technologies: Catching the Wave. *Harvard Business Review,* Jan-Feb 1995, pp.43-53.

S. Brin & L. Page (1998) The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*. **30**(1-7):107-117. Available here: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.96.2493&rep=rep1&type=pdf.

A. Byde (2003). Applying Evolutionary Game Theory to Auction Mechanism Design. In *Proceedings of the 2003 ACM Conference on E-Commerce*, pp.192-193. Also available as Hewlett-Packard Labs Technical Report HPL-2002-321, available from http://www.hpl.hp.com/techreports/2002/HPL-2002-321.pdf.

N. Carr (2008). *The Big Switch: Rewiring the World from Edison to Google.* W. W. Norton.

CFTC & SEC (2010b). *Findings Regarding the Market Events of May 6th, 2010.* Report of the staffs of the CFTC and SEC to the Joint Advisory Committee on Emerging Regulatory issues. September 30th, 2010. http://www.sec.gov/news/studies/2010/marketevents-report.pdf

C. Christensen (2003). *The Innovator's Dilemma*. Harper Collins.

D. Cliff (1997). *Minimal-Intelligence Agents for Bargaining Behaviours in Market Environments.* Hewlett-Packard Labs Tech Report HPL-97-91. Available from http://www.hpl.hp.com/techreports/97/HPL-97-91.html.

D. Cliff (1998). Evolutionary Optimization of Parameter Sets for Adaptive Software-Agent Traders in Continuous Double Auction Markets. Presented at the *Artificial Societies and Computational Markets Workshop (ASCMA98)*, May 1998. Available from http://www.hpl.hp.com/techrepropts/2001/HPL-2001-99.pdf.

D. Cliff (2003). Explorations in Evolutionary Design of Online Auction-Market Mechanisms. *J. Electronic Commerce Research & Applications.* **2**(2):162-175.

D. Cliff (2009). ZIP60: Further Explorations in the Evolutionary Design of Trader Agents and Online Auction-Market Mechanisms. *IEEE Transactions on Evolutionary Computation.* **13**(1):3-18.

D. Cliff (2010). *Remotely Managed Services and "Cloud Computing".* Emerging Technology for Learning Report, BECTA. http://www.cs.bris.ac.uk/home/dc/cliff_becta_clouds.pdf.

D. Cliff & L. Northrop (2011). *The Global Financial Markets: An Ultra-Large-Scale Systems Perspective*. Foresight Driver Review DR4.

P. Cornish, D. Livingstone, D. Clemente, & C. Yorke (2010). *On Cyber Warfare*. Chatham House Report, November 2010. Available from http://www.chathamhouse.org.uk.

N. Cristianini & J. Shawe-Taylor (2000). *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods.* Cambridge University Press.

G. Cugola & A. Margara (2011, forthcoming). "Processing Flows of Information: From Data Stream to Complex Event Processing" To appear in *ACM Computing Surveys*.

E. Daneels (2004). Disruptive Technology Reconsidered: A Critique and Research Agenda. *Journal of Product Innovation Managemen*t. **21**:246-258.

R. Das, J. Hanson, J. Kephart, & G. Tesauro (2001). Agent-Human Interactions in the Continuous Double Auction. *Proc. IJCAI* 01, pp.1169-1187.

M. De Luca & D. Cliff (2011a). Agent-Human Interactions in the Continuous Double Auction, Redux: Using the OpEx Lab-in-a-Box to Explore ZIP and GDX. *Proceedings of the 3rd International Conference on Agents and Artificial Intelligence (ICAART)*, Rome.

M. De Luca & D. Cliff (2011b). Agent-Human-Agent Auction Interactions: Adaptive-Aggressive Agents Dominate. To appear in: *Proceedings of the International Joint Conference on Artificial Intelligence, (IJCAI2011)*,

M. De Luca, C. Szostek, J. Cartlidge, & D. Cliff (2011, forthcoming). *Studies of interactions between human traders and algorithmic trading systems.* Foresight Driver Review DR13.

T. Demos (2011). Cyber-attack raises SEC Questions. *The Financial Times.* 9th Feb., 2011. http://www.ft.com/cms/s/0/c4cb1716-33eb-11e0-b1ed-00144feabdc0.html - axzz1HDMFwITW

H. Dreyfus (1981). From Micro-Worlds to Knowledge Representation: AI at an Impasse. In *Mind Design*, edited by J. Haugeland. MIT Press: pp.161-204.

H. Dreyfus (1992). *What Computers Still Can't Do: A Critique of Artificial Reason.* MIT Press.

D. Easley, M. Lopez de Prado, & M. O'Hara (2011). The Microstructure of the Flash Crash: Flow Toxicity, Liquidity Crashes and the Probability of Informed Trading. *The Journal of Portfolio Management*, **37**(2):118-128.

O. Etzion & P. Niblett (2011). *Event Processing in Action.* Manning.

D. Fensel, J. Hendler, H. Lierberman, & W. Wahlster, editors, (2003). *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential.* MIT Press.

J. Ghosh & R. Ramamoorthi (2010). *Bayesian Nonparametrics*. Springer.

S. Gjerstad & J. Dickhaut (1998). Price Formation in Double Acutions. *Games and Economic Behavior*, 22:1-29.

D. Goldberg (1987). *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley Professional.

P. Gomber, B. Arndt, M. Lutat, & T. Uhle (2011). *High Frequency Trading*. Technical Report, Goethe Universität & Deutsche Börse.

J. Grant & T. Demos (2011). Derivatives next step for Chi-X after BATS deal. *The Financial Times.* Feb. 17th, 2011.

G. Gregoriou, editor (2010). *The Handbook of Trading: Strategies for Navigating and Profiting from Currency, Bond, and Stock Markets.* McGraw-Hill.

J. Grossklags & C. Schmidt (2006). Software Agents and Market (In)Efficiency - A Human Trader Experiment, *IEEE Transactions on System, Man, and Cybernetics: Part C* (Special Issue on Game-theoretic Analysis & Simulation of Negotiation Agents) **36**(1):56-67.

J. Grossklags (2007) Experimental Economics and Experimental Computer Science: A Survey, *Workshop on Experimental Computer Science (ExpCS'07), ACM Federated Computer Research Conference (FCRC)*, San Diego, CA.

S. Gulati & W. Padgett (2003). *Parametric and Nonparametric Inference from Record-Breaking Data.* Springer.

G. Hallsmith & B. Lietaer, (2011). *Creating Wealth: Growing Local Economies with Local Currencies.* New Society Publishers.

S. Haykin (2008). *Neural Networks and Leaning Machines*. Third Edition. Pearson.

N. Hjort, C. Holmes, P. Müller & S. Walker (2010). *Bayesian Nonparametrics*. Cambridge University Press.

IBM (2006). *The Trader Is Dead: Long Live The Trader!* Report available from: http://www-935.ibm.com/services/us/index.wss/ibvstudy/imc/a1024121?cntxt=a1005266.

P. Jääskeläinen, C. de La Lama, P. Huerta, & J Takala, (2010). OpenCL-based design methodology for application-specific processors. In *Proc. 2010 International Conference on Embedded Computer Systems (SAMOS)* pp.223-230.

P. Janert (2011). *Data Analysis with Open-Source Tools*. O'Reilly.

J. Kagel & A. Roth (1997). *The Handbook of Experimental Economics.* Princeton University Press.

M. Kearns, A. Kulesza, & Y. Nevmyvaka (2010). Empirical Limitations on High Frequency Trading Profitability. Submitted to *Journal of Trading.* Available from http://www.cis.upenn.edu/~mkearns/papers/hft_arxiv.pdf.

T. Ladd, F. Jelezko, R. Laflamme, Y. Nakamura, C. Monroe, & J. O'Brien (2010). Quantum computers. *Nature*, **464**:45-53.

P. Larrañaga & J. Lozano (2001). *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation.* Springer.

D. Leinweber (2009) *Nerds on Wall Street: Math, Machines, and Wired Markets.* John Wiley & Sons, Inc.

J. Leyden (2011a). Nasdaq admits hackers planted malware on web portal. *The Register*, 7th Feb, 2011. http://www.theregister.co.uk/2011/02/07/nasdaq_malware_breach/.

J. Leyden (2011b). Tainted ads punt scareware to surfers on LSE and Myvue sites. *The Register*. 28 Feb 2011. http://www.theregister.co.uk/2011/02/28/tainted_ads_blight_uk_sites/

B. Lietaer (2002). *The Future of Money: Creating New Wealth, Work and a Wiser World*. Century.

R. Lowenstein (2002). *When Genius Failed: The Rise and Fall of Long-Term Capital Management.* Fourth Estate.

J. Lozano, P. Larrañaga, I. Inza, & E. Bengoetxea (2010). *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms.* Springer.

D. Luckham (2002). *The Power of Events: An introduction to complex event processing in distributed enterprise systems.* Addison-Wesley.

D. MacKay (2008). *Sustainable Energy – Without the Hot Air*. E-book downloadable from: http://www.withouthotair.com/download.html

A. Madhavan (2002). VWAP Strategies. *Transaction Performance: The Changing Face of Trading Investment Guide Series*, Institutional Investor Inc., pp.32-38.

C. Markides & D. Oyon (2010) What to do Against Disruptive Business Models (When and How to Play Two Games at Once). *MIT Sloan Management Review* **51**(4): 25-32.

D. May (2011) words transcribed from promotional video available at http://www.xmos.com/technology/xcore

S. McIntosh-Smith (2011) *From Multi-Core to Many-Core: The Next Important Trend in Computer Architectures.* A Knowledge Transfer Report from the London Mathematical Society and the Knowledge Transfer Network for Industrial Mathematics

M. Meerman, *et al*., (2011). *Money and Speed: Inside the Black Box.* Documentary produced by VPRO (Dutch public broadcaster), available as an iPad application. http://itunes.apple.com/us/app/money-speed-inside-black-box/id424796908?mt=8&ls=1#

G. Mitra *et al.* (2011, forthcoming). *Market Sentiment: Its Impact on Liquidity and Trading.* Foresight Driver Review DR8.

R. Miller (2001). *Experimental Economics: How We Can Build Better Financial Markets.* Wiley.

W. Moore, A. McCabe, & R. Urquhart, editors (1987). *Systolic Arrays.* IOP Publishing: Adam Hilger Books.

S. Mundy (2011) Birmingham Exchange Sputters. *The Financial Times,* 6th June, 2011. http://www.ft.com/cms/s/0/c422f918-8fd6-11e0-954d-00144feab49a.html#axzz1PdqwFSxQ

A. Munshi, B. Gaster, T. Mattson, J. Fung, & D. Ginsburg (2011). *OpenCL Programming Guide.* Addison-Wesley, forthcoming.

R. Narang (2009). *Inside the Black Box: The Simple Truth About Quantitative Trading.* Wiley Finance.

L. Neal (2000). How it all began: the monetary and financial architecture of Europe during the first global capital markets, 1648–1815. *Financial History Review,* **7**:117–140.

J. Nocera (2009). Risk Mismanagement. *The New York Times*, January 2nd, 2009. http://www.nytimes.com/2009/01/04/magazine/04risk-t.html?_r=1&pagewanted=1

G. Nuti, M. Mirghaemi, C. Yingsaeree, & P. Treleaven, (2011) Algorithmic Trading, *IEEE Computer*, pp.8-10. January 2011.

E. Oja (1982). A Simplified Neuron Model as a Principal Component Analyzer. *Journal of Mathematical Biology,* **15**:267-273.

M. Pett (1997). *Nonparametric Statistics for Health Care Research: Statistics for small samples and unusual distributions*. Sage.

S. Phelps, S. Parsons, P. McBurney, & E. Sklar (2002). Co-evolutionary mechanism design: A preliminary report. In J. Padget, O. Shehory, D. Parkes, N. Sadeh, & W. Walsh (editors) *Agent-Mediated Electronic Commerce IV: Designing Mechanisms and Systems*, pp.123–143. Springer.

S. Phelps, P. McBurney, & S. Parsons (2009). Evolutionary Mechanism Design: A Review. *Autonomous Agents and Multi-Agent Systems,* **21**(2):237-264.

A. Pole (2007). *Statistical Arbitrage: Algorithmic Trading Insights and Techniques.* Wiley Finance.

D. Richardson (2010). Filling the Light Pipe. *Science*, **330**(6002):327-328.

J. Sanders & E. Kandrot (2010). *CUDA by Example: An Introduction to General-Purpose GPU Programming.* Addison-Wesley.

SEC & BCG (2011). *U.S. Securities and Exchange Commission, Organizational Study and Reform.* Report from Boston Consulting Group, published by SEC, March 10th 2011.

S Siegel & N. Castellan (1988). *Nonparametric Statistics for the Behavioral Sciences.* Second edition. New York, NY: McGraw-Hill.

V. Smith (1991). *Papers in Experimental Economics*. Cambridge University Press.

V. Smith (2000). *Bargaining and Market Behavior: Essays in Experimental Economics*. Cambridge University Press.

Sophos (2006). News story available at http://www.sophos.com/pressoffice/news/articles/2006/02/russiantrading.html.

P. Stafford, J. Grant, & T. Demos (2011). Hacking fears raised by Nasdaq OMX attack. *The Financial Times*, 7th Feb 2011. http://www.ft.com/cms/s/0/0638d37a-32fa-11e0-9a61-00144feabdc0.html#axzz1HDMFwITW.

C. Steiner (2010). Wall Street's Speed War. *Forbes Magazine*, September 27th 2010. http://www.forbes.com/forbes/2010/0927/outfront-netscape-jim-barksdale-daniel-spivey-wall-street-speed-war.html

N. Stephenson (1996). Mother Earth Mother Board. *Wired*, 4(12), December 2006. http://www.wired.com/wired/archive/4.12/ffglass.html.

C. Stephens & H. Waelbroeck (2009). Algorithm Switching: Co-Adaptation in the Market Ecology. *The Journal of Trading,* **4**(3):1-15.

A. Thompson (1998). *Hardware Evolution: Automatic Design of Electronic Circuits in Reconfigurable Hardware by Artificial Evolution.* Springer.

H. Taylor, A. Yochem, L. Phillips, & F. Martinez (2009). *Event-Driven Architecture.* Addison-Wesley.

L. Van Valen (1973). A New Evolutionary Law. *Evolutionary Theory,* **1**:1-30.

L. Valiant (1984). A Theory of the Learnable. *Communications of the ACM.* **27**(11): 1134-1142.

R. Weiss, S. Basu, S. Hooshngi, A. Kalmbach, D. Karig, R. Mehreja, & I. Netravali (2003). Genetic circuit building blocks for cellular computation, communications, and signal processing. *Natural Computing,* **2**(1).

T. White (2010). *Hadoop: The Ultimate Guide*. O'Reilly.

A. Wissner-Gross & C. Freer (2010). Relativistic Statistical Arbitrage. *Physical Review E*. **82:**056104.

D. Wolpert (1994). The Relationship Between PAC, the Statistical Physics Framework, the Bayesian Framework, and the VC Framework. In D. Wolpert (editor), *The Mathematics of Generalization*. Addison-Wesley.

L. Yu, H. Chen, S. Wang, & K. Lai (2009). Evolving Least Squares Support Vector Machines for Stock Market Trend Mining. *IEEE Transactions on Evolutionary Computation.* **13**(1):87-102.

K. Zauner (2005a). From Prescriptive Programming of Solid-state Devices to Orchestrated Self-organisation of Informed Matter. In*: Unconventional Programming Paradigms: International Workshop UPP 2004,* Revised Selected and Invited Papers, LNCS, **3566**: 47-55, Springer.

K. Zauner (2005b). Information Science Meets the Material World. In: *Towards 2020 Science*, 30 June - 2 July 2005, Venice.

K. Zauner (2005c). Molecular Information Technology. *Critical Reviews in Solid State and Material Sciences,* **30**(1):33-69.

Foresight
1 Victoria Street
London SW1H 0ET
www.foresight.gov.uk
URN 11/1222